JATISKOM
Jurnal Aplikasi Teknologi Informasi dan Sains Komputer
e-ISSN : 3089-8668
https://journal.uinmataram.ac.id/index.php/jatiskom

Home Security System Based on Arduino Uno Using SMS

Yudi Ahsana Maswa[1], Lalu Puji Indra Kharisma[2], Arief Taufikurrahman[3], Sri Andani[4]

[1] STMIK Syaikh Zainuddin NW; yudiahsana@gmail.com
[2] STMIK Syaikh Zainuddin NW; lalupuji@gmail.com
[3] Universitas Islam Negeri Mataram; arief@uinmataram.ac.id
[4] Program Studi Teknologi Informasi Universitas Islam Negeri Mataram; andanyzzz238@gmail.com

* Korespondensi: lalupuji@gmail.com

Abstract Security is a crucial aspect for everyone. Homes that are frequently left unattended by their owners can become targets for thieves to carry out their actions. Therefore, an effective home security system is needed to enable homeowners to receive information if their house is broken into by a thief. The author has developed a home security system utilizing technology with an Infrared sensor and GSM module as an SMS alert information system, where all components are processed in an Arduino Uno microcontroller.

The research methodology used includes several stages: data collection (literature review, observation, and experimentation) and software development using the prototype method (communication, quick planning, quick design modeling, prototype construction, deployment delivery, and feedback).

The home security system works by using sensors placed at entry points, such as doors. This system automatically sends an SMS to the homeowner's mobile phone, regardless of their location.

Keywords: *Security system, Arduino Uno, Infrared sensor*

1. Introduction

The industrial world plays a very important role in the development of technology. Therefore, humans continue to innovate to create appropriate technologies that make life easier. However, it cannot be denied that alongside the positive impacts of increasingly modern technology, there is also a rise in criminal activities in various regions. In Indonesia, in particular, the crime rate has been increasing year by year in recent years, with a growing variety of criminal acts. To prevent crimes, efforts are needed, one of which is by utilizing technology. Regarding home theft, home security technology is necessary, such as developing a home security system using technology.

A house is one of the basic needs of human life. It serves as a shelter from all kinds of weather and as a place for growth, development, and gathering of a small human community—the family. Every family living in their home deserves security and comfort. People use various methods to secure valuable items stored in a room within their house, such as using a padlock or hiring a security guard to protect the room. Protection against theft is the most desired aspect for everyone. However, safeguarding against theft is one of the most challenging tasks and is not easy to accomplish due to the limitations of human senses.

A home security system that uses an SMS-based microcontroller allows users to control the home security system remotely by sending SMS messages. This is very useful if users want to check the condition of their home while they are away. The system's ability to send important data to users via SMS makes it valuable for monitoring the home's surroundings, especially when the house is left unattended for a long period. Additionally, it enhances home security by directly notifying the homeowner of suspicious situations or any attempts at theft or vandalism.

Home security systems that are not yet optimal sometimes have vulnerabilities that can be exploited by intruders. Therefore, the author plans to develop a security system to prevent theft inside the house. Essentially, this tool is a security system consisting of a microcontroller and several electronic components. The electronic components used include an Arduino Uno, an infrared sensor, a GSM module, and a buzzer. The system will also be equipped with an SMS notification service to inform the homeowner. The infrared sensor will be placed on the house window. If someone enters through the window and the sensor detects movement, the homeowner will receive an SMS. If the SMS is not responded to, the alarm (buzzer) will activate simultaneously. In this project, the author aims to develop a security system using infrared sensor technology as the input, Arduino Uno as the microcontroller, and SMS notifications as the output.

Based on the aforementioned problem description, the author is interested in selecting the title for the final project: "MICROCONTROLLER-BASED HOME SECURITY SYSTEM USING SMS."

2. Materials and Methods

In preparing this research, the author used several stages or methods, namely:

1. Data Collection Stage
    A. Literature Method

    The author collects data by reading and studying relevant literature such as books, journals, and scientific articles.

    B. Observation Method

    The author conducts direct observations to determine the necessary data for creating a prototype model so the device can function.

    C. Experimental Method

    In this method, the author conducts trials and implements the newly created device, corrects any errors that occur, and records the results of the experiments.

2. Software Development Stage

In preparing this final project, the author uses the Prototype Method for the software development model. According to Pressman (2010), the Prototype Method has several stages, namely:

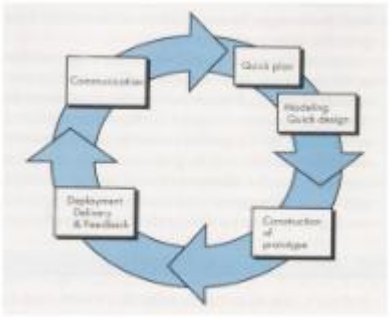

Figure 1.1 Prototype Model Process

3. Communication

This step involves analyzing the system requirements, or analyzing all the needs of the project. This phase is used to identify the scope of information, the required functions, and the performance capabilities to be achieved

4. Quick Plan

This step involves the process of designing the interface based on the completed requirement analysis. The design is created in such a way that users find it easy to use.

5. Modeling Quick Design

This process focuses on presenting the prototype creation, designing a temporary model of a miniature house as an illustration for a home security system.

6. Construction of Prototype

The Construction of Prototype is the process of writing the code. Once completed, testing must be conducted immediately to minimize coding errors. This phase determines whether the built prototype meets the expected results. If it meets the expectations, the next step will be taken. However, if it does not meet expectations, the prototype is revised by repeating the previous steps.

7. Deployment, Delivery & Feedback

This stage can be considered the final phase in the creation of a project. The process includes several activities, such as correcting errors in the prototype, improving the system units that have been implemented, and enhancing the system services for the home.

In the prototype model, the tool that has been created cannot be considered perfect right away, as there is still the possibility of changes being made by the developer or the user to the system that has been designed. Therefore, if there are any changes to the system, the design process must start from the beginning, which is the communication process.

3. Results

At this stage, the author will analyze the system that will be built later. The analysis is as follows:

1. System Weakness Analysis
   Many thefts occur when the homeowner is traveling and the house is left for a long time, of course this makes it easier for thieves to make the house a target for theft.
2. System Needs Analysis
   System needs analysis is a process stage of all types of needs used in the tool development process, including hardware needs and software needs:
   a) Hardware Requirements
      1. Asus X455LF Laptop (Windows 8.1 64-bit operating system), Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz (4 CPUs) Processor, ~2.2GHz, 4096MB RAM Memory
      2. Arduino Uno which is useful for storing all commands or functions in the home security system.
      3. The Infrared Sensor is a component that the author uses to detect movement from doors and windows.
      4. The Buzzer is an electronic component that functions to convert electrical vibrations into sound vibrations.
      5. GSM Module (SIM900A V4.0) is a component used for data communication between cellular network systems.
      6. SIM card is a component used as a notification connection to the user's cellphone.
   b) Software Requirements
      1. Arduino IDE is a software application that the author uses to process and translate all data into the system using the C language.
      2. Windows 10 Operating System For a minimum operating system of Windows 10 to use Arduino IDE.
      3. System Feasibility Analysis

This sub-chapter will discuss the analysis of Technology feasibility, Legal feasibility and Operational feasibility.

1. Technology Feasibility
   In this case, the author created a home security system via SMS (Short Message Service) notification using Arduino Uno infrared sensor technology which will later be used to avoid or minimize the occurrence of theft in the house.

2. Legal Feasibility
   In developing a device or system based on Arduino Uno or microcontroller is an open source development of both software and hardware. This means that this device is free to use, copy, distribute, and develop by anyone without having to pay

or ask for official permission from the company so that there is no violation of the law.

3. Operational feasibility

   In this case, the author plans to create a home security prototype via SMS using Arduino Uno pir sensor technology that can be easily operated and understood by the homeowner

The author makes a design for the system to be built. The design of the system is as follows:

1. Process design
   a. Home Security System Block Diagram

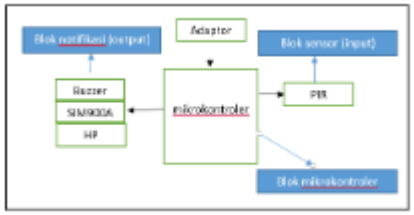

Table 3.1 Home Security System Block

The Home Security System Block Diagram as shown in Figure 4 can be explained as follows:

1. The block diagram of this home security system works from PLN electricity, then forwarded to the adapter electricity to change the voltage from PLN to 20 V, after which it is used to supply power to the Arduino Uno microcontroller.
2. All components used use power from the microcontroller in their operation.
3. When the sensor detects movement, the signal will be forwarded to the microcontroller
4. The Arduino microcontroller will process the signal received from the infrared sensor and will determine whether there is an indication of danger or not. If there is an indication of danger, the Arduino will send a command to the SIM900A module.
5. The SIM900A module is used to send SMS notifications to the specified number. After receiving a command from the Arduino, the SIM900A module will send an SMS notification to the previously set number.
6. A cellphone or mobile phone is used to receive SMS notifications from the SIM900A module.
7. When the Arduino detects a danger indication, the buzzer will sound as an additional warning.

b. Flowchart

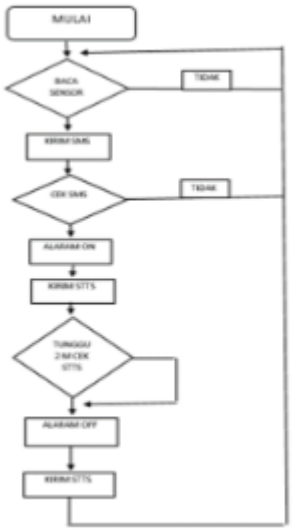

Table 3.1 Flowchsrd table

1. The system starts by checking the condition of the infrared sensor. If the infrared sensor detects movement, the system will check the condition of the SIM900A.
2. If the SIM900A is ready, the system will send an SMS to the specified number
3. After sending the SMS, the system will activate the buzzer as an additional alarm.
4. The system will wait for a few seconds to ensure that no more movement is detected by the infrared sensor.
5. The homeowner will receive an SMS message. The homeowner can check his cellphone to see if there is any indication of danger.

c. Prototype creation

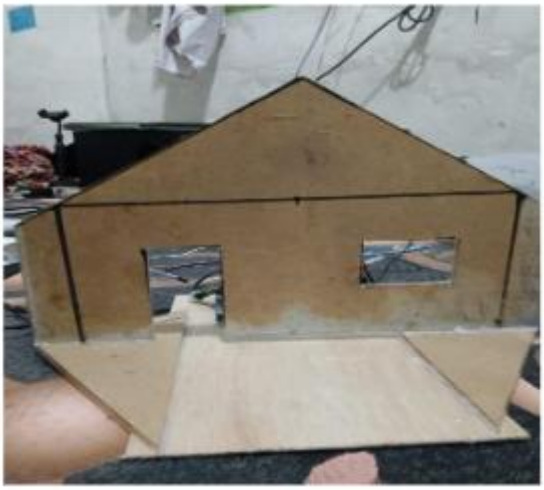

Figure 3.9 Making a miniature house prototype

Figure 3.9 is the final stage, namely making a miniature house prototype with acrylic.

4. Discussion

1. System Implementation
   A. Hardware Implementation

Hardware implementation is the process of turning an electronic design into a real-world device that will be used as hardware for the home security system based on a microcontroller and SMS. This process involves designing the Arduino device and integrating it with supporting components.

For the Arduino device design, it is necessary to outline and determine which pins will be used for input and output connections. The design of the Arduino circuit schematic was created using a software tool called **Fritzing**, as shown in **Figure 4.1**. In the design process, jumper wires are drawn from the output pins and connected to the input pins of the predetermined components.

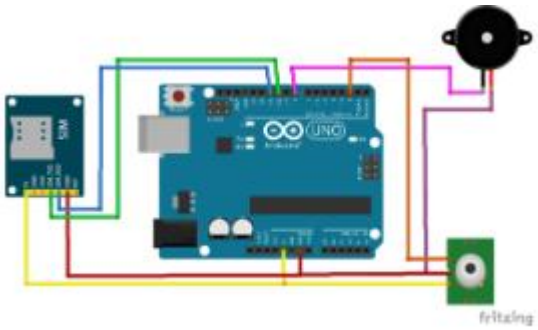

Figure 4.1 Device Design

After the Arduino device design has its respective connections as shown in Figure 4.1, this design will then be implemented on the Arduino, integrated with other components.

The first circuit in creating a home security system integrated with the Arduino microcontroller is to assemble several voltage supply components. These components include an infrared sensor, a buzzer, a SIM 900A module, and an Arduino Uno microcontroller. This can be seen in Figure 4.3.

2. Software Implementation

In the creation of a home security system through SMS notifications and phone calls using Arduino Uno with infrared sensor technology, the author uses the Arduino IDE application as the program development environment. In this subsection, the author will explain how to use the Arduino IDE as follows:

a. Open the Arduino IDE, and a sketch window will appear as shown in Figure 23 below:

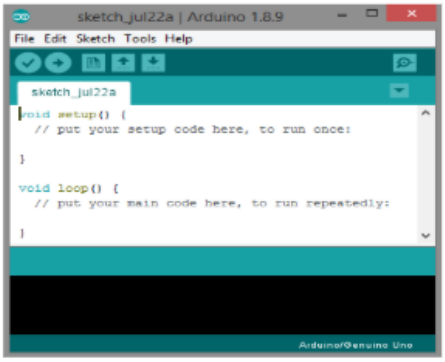

Figure 4.1 Arduino IDE Application

b. There are various types of boards available in the Arduino IDE tools. Since the author is using the Arduino Uno board, select the Arduino Uno board

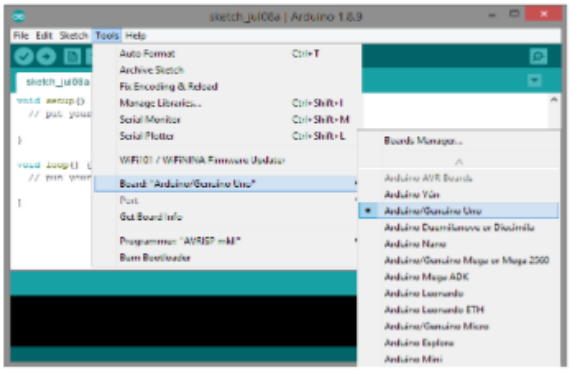

Figure 4.2 Arduino IDE Board Settings

c. The figure shows the settings for the port for the Arduino board.

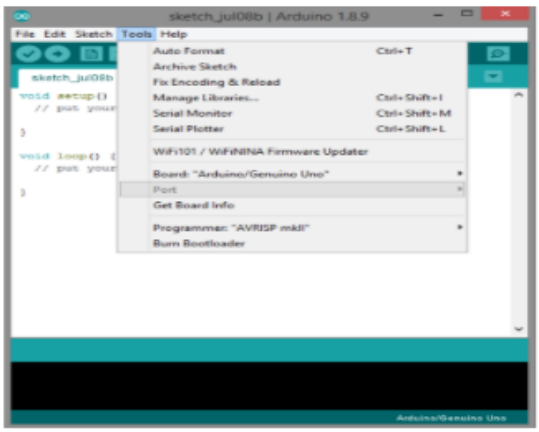

Figure 4.3 Arduino Board Port Settings

d. Next, create the source code, or commonly called a sketch, in the Arduino IDE. The file saved in the Arduino IDE has a .ino extension. Once saved, verify/compile it to the Arduino Uno board as shown in Figure 26 below.

Figure 4.4 How to Write Code in the Arduino IDE Application

3. Function of the Program Source Code

In this subsection, the author will explain the source code used in the development of this home security system program.

a. In the first line, `#include <SoftwareSerial.h>` is a preprocessor directive used to include the `SoftwareSerial` library. This library is used to establish serial communication with other devices through specified digital pins. It is commonly used for communication with devices such as GSM modules, Bluetooth, or other devices that use a serial protocol.

```
#include <SoftwareSerial.h>

SoftwareSerial SIM900A(10, 11);
const int irPin = 2; // Pin number for the infrared sensor
const int buzzerPin = 3; // Pin number for the buzzer sensor
```

Figure 4.5 Serial Communication

The second line, `SoftwareSerial SIM900A(10, 11);`, declares an object `SIM900A` of the `SoftwareSerial` class. This object is used to communicate with the SIM900A device through digital pins 10 (TX) and 11 (RX) on the Arduino. In this case, the SIM900A is a GSM module used for cellular communication.

Next, `const int irPin = 2;` defines the variable `irPin` with the value 2. This variable is used to store the digital pin number that is used to connect the infrared sensor.

Lastly, `const int buzzerPin = 3;` defines the variable `buzzerPin` with the value 3. This variable is used to store the digital pin number that is used to connect the buzzer (sound generator).

b. The code below is the implementation of the `setup()` function in C++ for Arduino. The `setup()` function is used for initial initialization and configuration settings on the Arduino device before the main program runs.

```
void setup()
{
  SIM900A.begin(9600);   // Setting the baud rate of GSM Module
  Serial.begin(9600);    // Setting the baud rate of Serial Monitor (Arduino)
  Serial.println("SIM900A Ready");
  pinMode(irPin, INPUT); // Initialize infrared sensor pin as input
  pinMode(buzzerPin, OUTPUT); // Initialize buzzer sensor pin as output
  delay(100);
  Serial.println("Sending message...");
  SendMessage();
}
```

Figure 4.6 Implementation of the `setup()` Function

In the `setup()` block, several commands are executed as preparation steps before the main program runs:ss

1) `SIM900A.begin(9600);` sets the baud rate (9600 bps) for communication with the GSM SIM900A module. The `begin()` function is used to start serial communication with the specified baud rate.
2) `Serial.begin(9600);` sets the baud rate (9600 bps) for communication with the Serial Monitor connected to the Arduino. The `begin()` function is used to initiate serial communication with the specified baud rate.
3) `Serial.println("SIM900A Ready");` sends the text "SIM900A Ready" to the Serial Monitor. The `println()` function is used to print text and move to the next line.
4) `pinMode(irPin, INPUT);` configures the `irPin` as an input. The `pinMode()` function is used to set the mode of a digital pin, in this case, as input.
5) `pinMode(buzzerPin, OUTPUT);` configures the `buzzerPin` as an output. The `pinMode()` function is used to set the mode of a digital pin, in this case, as output.
6) `delay(100);` creates a delay of 100 milliseconds. The `delay()` function is used to introduce a pause for a specified duration.
7) `Serial.println("Sending message...");` sends the text "Sending message..." to the Serial Monitor.

8) `SendMessage();` calls the `SendMessage()` function. This function is not shown in the provided code snippet but may be defined elsewhere in the program. It is responsible for sending messages through the GSM module.

c. The code below this explanation is an implementation of the `loop()` function in C++ for Arduino. The `loop()` function is executed repeatedly in an endless cycle after the `setup()` function has been completed. It is used to run the main program and perform repetitive tasks based on the defined logic.

```
void loop()
{
  if (SIM900A.available() > 0)
    Serial.write(SIM900A.read());

  int irStatus = digitalRead(irPin);
  if (irStatus == LOW)
  {
    Serial.println("Object detected!");
    SendMessage();
    activateBuzzer();
    delay(5000); // Delay to avoid multiple messages within a short time
    deactivateBuzzer();
  }
  else
  {
    Serial.println("No object detected.");
    delay(1000); // Delay when no object is detected
  }
}
```

Figure 4.7: The '`loop()`' Function

In the `loop()` block, several commands are executed repeatedly:

1. `if (SIM900A.available() > 0)` is a condition that checks whether there is data available to be read from the SIM900A GSM module. The `available()` function is used to get the number of bytes available to read from the `SIM900A` object. If data is available, the code block inside the if statement will be executed.
2. `Serial.write(SIM900A.read())` writes the data received from the SIM900A GSM module to the Serial Monitor. The `read()` function is used to read a byte of data from the `SIM900A` object, and the `write()` function is used to send that byte to the Serial Monitor.
3. `int irStatus = digitalRead(irPin)` reads the status (HIGH or LOW) from the `irPin`, which is connected to the infrared sensor. The `digitalRead()` function is used to read the digital value on the specified pin.
4. `if (irStatus == LOW)` is a condition that checks whether the infrared sensor detects an object. If the value of `irStatus` is LOW (indicating an object is detected), the code block inside the if statement will be executed.
5. `Serial.println("Object detected!")` prints the text "Object detected!" to the Serial Monitor.

6. `SendMessage()` calls the `SendMessage()` function, which is responsible for sending a message through the GSM module.
7. `activateBuzzer()` calls the `activateBuzzer()` function, which is responsible for activating the buzzer (sound generator).
8. `delay(5000)` introduces a delay of 5000 milliseconds (5 seconds) to prevent repeated message sending within a short time.
9. `deactivateBuzzer()` calls the `deactivateBuzzer()` function, which is responsible for deactivating the buzzer.
10. If the condition `irStatus` is not met (i.e., `irStatus` is HIGH, meaning no object is detected), the code block inside the else statement will be executed.
11. `Serial.println("No object detected.")` prints the text "No object detected." to the Serial Monitor.
12. `delay(1000)` introduces a delay of 1000 milliseconds (1 second) when no object is detected, before starting the next iteration of the loop.

d. The `SendMessage()` function is used to send a text message via the SIM900A GSM module.

```
void SendMessage()
{
  SIM900A.println("AT+CMGF=1");    // Sets the GSM Module in Text Mode
  delay(1000);
  SIM900A.println("AT+CMGS=\"+6287767094081\""); // Mobile phone number to send message
  delay(1000);
  SIM900A.println("Object detected!"); // Message content
  delay(100);
  SIM900A.println((char)26); // ASCII code of CTRL+Z
  delay(5000); // Delay for the message to be sent successfully
  Serial.println("Message has been sent");
}
```

Figure 4.8: `SendMessage()` Function

Below is an explanation of each command within the `SendMessage()` function:

1. **`SIM900A.println("AT+CMGF=1")`**
   Sends an AT command to the SIM900A GSM module to set it to text mode (`CMGF=1`). This configures the module to send and receive text messages.
2. **`delay(1000)`**
   Introduces a delay of 1000 milliseconds (1 second) to allow the SIM900A GSM module sufficient time to process the command before the next command is sent.
3. **`SIM900A.println("AT+CMGS=\"+6287767094081\"")`**
   Sends an AT command to the SIM900A GSM module to set the recipient's phone

number. In this example, the target phone number is `+6287767094081`. Note that the phone number must be enclosed in double quotes and start with the `+` character.

4. **`delay(1000)`**
   Introduces a delay of 1000 milliseconds (1 second) before sending the message content.
5. **`SIM900A.println("Object detected!")`**
   Sends the text message "Object detected!" to the SIM900A GSM module. This message will be sent to the previously specified phone number.
6. **`delay(100)`**
   Introduces a delay of 100 milliseconds (0.1 seconds) before sending the Ctrl+Z character (ASCII code 26), which marks the end of the message.
7. **`SIM900A.println((char)26)`**
   Sends the Ctrl+Z character (ASCII code 26) to the SIM900A GSM module. This signifies the end of the message and triggers its transmission to the recipient.
8. **`delay(5000)`**
   Introduces a delay of 5000 milliseconds (5 seconds) to ensure the message has been successfully sent before proceeding to the next step. This delay can be adjusted depending on network conditions and message transmission speed.
9. **`Serial.println("Message has been sent")`**
   Prints the text "Message has been sent" to the Serial Monitor as an indication that the message has been successfully delivered.

Using the `SendMessage()` function, you can send a text message with a specified content to the desired phone number via the SIM900A GSM module.

Additionally:

e. The **`activateBuzzer()`** and **`deactivateBuzzer()`** functions are used to control the buzzer (sound generator) connected to the `buzzerPin`.

```
void activateBuzzer()
{
  digitalWrite(buzzerPin, LOW); // Turn on the buzzer
}

void deactivateBuzzer()
{
  digitalWrite(buzzerPin, HIGH); // Turn off the buzzer
}
```

Figure 4.9: The `activateBuzzer()` Function

1. The `activateBuzzer()` function is used to activate the buzzer. Within this function, the command `digitalWrite(buzzerPin, LOW)` is executed, which sets the `buzzerPin` to LOW. In this context, setting the pin to LOW activates the buzzer.

2. The `deactivateBuzzer()` function is used to deactivate the buzzer. Within this function, the command `digitalWrite(buzzerPin, HIGH)` is executed, which sets the `buzzerPin` to HIGH. In this context, setting the pin to HIGH deactivates the buzzer.

By using these two functions, you can control the buzzer by calling `activateBuzzer()` to turn it on and `deactivateBuzzer()` to turn it off.

3. Testing the Device

In this subsection, the author conducts several tests on the system for home security through SMS notifications and phone calls using Arduino Uno with infrared sensor technology. To ensure that the device functions as expected, testing is required for the program that has been developed.

No	**Test Functional Requirements**	**Input Values**	**Results**	**Status**
1	Connection Test to Arduino Uno	Set Detected *Port*	Detected *Port*	Success
2	*Infrared* Sensor Test	ing the Door	*Infrared* Sensor Light Activation	Success
3	GSM *Module* Test	ing the Door	Sending SMS to *User*	Success`

5. Conclusions

Based on the previous chapters such as analysis, system design, system implementation, and discussion, the author can draw the following conclusions:

1. This Home Security System through SMS Notification and Using Infrared Sensor Technology with Arduino Uno is built using the Arduino Uno board and the GSM module (SIM900A V4.0) as the authentication tool for the homeowner.

2. The security system is designed to prevent theft inside the house or the use of duplicate keys when the homeowner is away or out of their direct supervision.
3. The operation of this home security system is as follows: when the sensor placed on the door detects movement, the system will automatically send an SMS to the homeowner's phone, no matter where they are.

References

[1] Dwi Yuny Sylfania (2019). Home Security System Prototype Using a Combination of Sensors and SMS Gateway

[2] Purnomo, D. (2017). Prototyping Model in Information System Development

[3] Asad, Muhammad Ridwan. (2015). Automatic Home Door Security System via SMS Based on ATMega328P Microcontroller. Journal of Technology and Computer Systems, 1-7.

[4] Kadir, Abdul. (2016). Scratch for Arduino (S4A). ANDI, Yogyakarta.

[5] Kurniawan, Muhammad Irfan. (2018). Home Security System Based on Raspberry Pi and Telegram Messenger. ELKOMIKA, 1-15.

[6] Pressman, R. S. (2010). Software Engineering: A Practitioner's Approach. Yogyakarta: Andi Publisher.

[7] Informatics Journal of Merdeka Pasuruan, 54-61.

[8] Ramadhan, Ade Surya. (2016). Design and Build of Home Security System Based on Arduino Mega 2560. Techno.COM, 117-124.

[9] Rerungan, J., Nugraha, D. W., & Anshori, Y. (2014). Automatic Door Security System Using Radio Frequency Identification (RFID) Tag Card and Personal Identification Number (PIN) Based on AVR Atmega 128 Microcontroller. ISSN 2356-4792.