CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

Fire Detection using IoT and GPS using Arduino IDE for programming. Here are three designs of Fire Detection using IoT and GPS project using Arduino Uno, among others:

1. Fire Sensor KY-026
2. GPS Ublox Neo-7M
3. ESP8266

Illustration 5.1: Prototype
5.1.1 Fire Sensor KY-026

Here are the results of the Fire Sensor-KY026 design on Arduino Uno and breadboard:

![Illustration 5.2: Fire Sensor](image)

Fire Sensor KY-026 uses a positive voltage of 5 volts which is then connected from Arduino Uno 5 volts at the foot pin above D0. This sensor also requires a negative voltage obtained from Arduino Uno and then connected to the foot of the pin that there is a description of G above the foot pin +. The jumper cable embedded in A0 is connected to the Arduino Uno on the pins between A0 and A5. What is meant by A0 is analog, and here Arduino Uno provides five analog pins which have a voltage of 5 volts which has a value of 0 – 1023.

5.1.2 GPS Ublox Neo-7M

Here are the results of the Ublox Neo-7M GPS design on Arduino Uno and breadboard:

![Illustration 5.3: GPS Ublox Neo-7M](image)
GPS Ublox Neo-7M uses a positive current voltage of 3.3 volts so connect a jumper cable on the VCC to a breadboard that has been given 3.3 volts to power the GPS, and also uses a negative voltage on the GND pin of the GPS module. Here the difference is TX and RX on the GPS module. RX alias Receiver is connected to TX pin alias Transceiver to transmit data to arduino so that it can do Serial.read. Instead, the TX pin is connected to the RX arduino to receive data from the arduino so that it can perform Serial.print. But to note is when connecting arduino to computer and upload sketch program, unplug the TX pin to RX which is at pin 0 in Arduino Uno thus avoiding the intrusion of communication between arduino and computer.

5.1.3 ESP8266

Here are the results of ESP8266 design on Arduino Uno and breadboard:

On ESP8266 requires the same voltage with GPS Nelo-7M that is equal to 3.3 volts. ESP8266 also requires RX and TX as well as GPS. However, in Arduino Uno only has serial hardware one pair only. So ESP8266 uses serial software so it can communicate between Arduino and ESP8266. Serial software is recommended to be used in only one module, because for two serial software is not effective since only one serial software will work. Here RX and TX on ESP8266 are connected on pins 9 and 8 as communication between ESP8266 and Arduino Uno.
Here are the obvious details of the test results that have been done starting from Fire sensors to detect the presence of fire until sms delivery:

1. Here when Arduino Uno is turned on, then the first working is ESP8266 to get internet connection which its usage to send data to thingspeak. Once connected to the internet, the Neo-7M GPS Ublox takes about 10 to 30 seconds to connect to the satellite signal depending on the location of whether many buildings or walls. When it is connected to satellite; the program will show the latitude and longitude of the location.

Illustration 5.5: Google Maps
Satellite will continue to work to locate the point of the house until the KY-026 flame sensor working on infrared light in the 760 nm - 1100 nm wavelength range with detection distance of less than 1 meter detects a fire then latitude, longitude, and flame sensors will be sent to thingspeak.

```
GET /update?api_key=0AZUKC23B5866FWM6field1=48&field2=7.061151&field3=110.431020
```

Illustration 5.6: GPS in Arduino
Delivery of data to thingspeak requires a lag time of approximately 20 seconds to actually have data into the database thingspeak.

Here PHP will retrieve data by using file_get_contents on thingspeak with results equal to 1 to retrieve the most recent data entries 154. Then split into array to retrieve data entries, latitude, and longitude.

```
```

Illustration 5.7: Thingspeak

Illustration 5.8: PHP

Latitude and Longitude data will then be calculated using the haversine formula method already discussed earlier in chapter 3 to produce the last result that is 4.663km.
In performing the Haversine calculation process, first enter the latitude and longitude data that have been taken from thingspeak in the $latTo and $lonTo variables. Here $latFrom and $lonFrom have provided five firefighting data.

Illustration 5.9: Haversine Formula

Illustration 5.10: PHP Code
Can be seen on lines 17, 18, 23, and 26 is to change the point of latitude and longitude to radian first.

```php
$latDelta = $latTo - $latFrom;
echo "LatDelta: ". $latDelta;
echo "<br/>
$lonDelta = $lonTo - $lonFrom;
echo "LonDelta: ". $lonDelta;
```

Illustration 5.11: PHP Code (2)

Then do the calculation of the reduction of the difference between the two points and stored on $latDelta and $lonDelta.

Obtained angle from the calculation which will then multiplied by $earthRadius that has been determined that is equal to 6371. Then the results are stored on the variable $hitungan2.

```php
$sin = 2 * asin(sin($latDelta / 2) * cos($latFrom) * cos($latTo) * pow(sin($lonDelta / 2), 2));
```

Illustration 5.12: PHP Code (3)

```php
$sin = sin($sin) * $earthRadius; //mengambil dalam bentuk array ke anu
 echo "Hasil: ", $sin;
```

Illustration 5.13: Earth Radius
Here five firefighting data will be calculated up to the fourth index because the first firefighter is counted as a zero index. Then from index to 0 to 4 compared where the shortest distance. Provided temp data greater than the distance calculation that will be used to compare data on index to 0. The temp data will be filled by the variable data index to 0 then will be compared to find the shortest distance and will be stored on the distance variables.

Illustration 5.14: PHP Code (4)

Previous data entries are used to avoid false alarms on sms gateway gammu to send sms because it is not possible php will send the same data on the same entry for the calculation. To avoid that happening, before doing insert data thing done first is to do select database at entry_id. Then comparing the entry_id in the database there is an existing entry_id, if different it will run the insert on the database because the result is NULL.
In the gateway gateway sms, which need to be entered only on the outbox of the form TextDecoded, Destination Number, and CreatorID. Here used simcard Indosat IM3 to send sms gateway.

Illustration 5.16: PHP Code (5)

Illustration 5.17: Gammu Outbox

Illustration 5.18: Gammu Sentitem
The data in the outbox table will move to the sentitems table if the sms has been successfully sent on the targeted phone.

Here is the form of sms submissions on the phone that has been addressed14.K1.0001

Illustration 5.19: SMS
5.2 Testing

Here are tests on fire sensors KY-026, ESP8266, GPS Neo-7M Ublox, PHP using haversine calculation, and Wavecom sms gateway gammu.

There are two tests on this project, including:

5.2.1 Performance Test

In the performance test performed five times on the fire sensor KY-026 is to measure the level of sensitivity or threshold by using wax and lighter at a predetermined distance.

1. Test sensitivity of Fire Sensor KY-026 using candle

<table>
<thead>
<tr>
<th>No</th>
<th>Value Fire Sensor (Threshold)</th>
<th>Jarak (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30 ~ 35</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>37 ~ 45</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>48 ~ 50</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>53 ~ 62</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>70 ~ 77</td>
<td>75</td>
</tr>
</tbody>
</table>

Table 5.1: Fire Sensor Candle

2. Fire Sensor KY-026 using lighter

<table>
<thead>
<tr>
<th>No</th>
<th>Value Fire Sensor</th>
<th>Jarak (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44 ~ 52</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>79 ~ 157</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>381 ~ 515</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>669 ~ 880</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>886 ~ 1015</td>
<td>75</td>
</tr>
</tbody>
</table>

Table 5.2: Fire Sensor Lighter

From the experiment results it can be seen that the indicator led light on the flame sensor when it detects a flame on the lighter flicker than when it detects a flame on the candle. When detecting a candle flame, the indicator's indicator light on the flame sensor always lights red at a distance of 15 cm to 75 cm. To avoid false alarm from the lighter, the threshold value of the fire sensor to be used
is 80 because the sensitivity level of the fire sensors is so sensitive to detect the presence of fire.

3. GPS Ublox Neo-7M accurate position

In the performance test will be tested five times to check the accuracy level of Ublox Neo-7M GPS module to determine the point of location of latitude and longitude. Which will then be compared with google maps how big errors are obtained.

<table>
<thead>
<tr>
<th>No.</th>
<th>Start up</th>
<th>Latitude (GPS)</th>
<th>Longitude (GPS)</th>
<th>Latitude (Google Maps)</th>
<th>Longitude (Google Maps)</th>
<th>Errors Latitude</th>
<th>Errors Longitude</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>-7.055172</td>
<td>110.429748</td>
<td>-7.055228</td>
<td>110.429717</td>
<td>5.59E-6</td>
<td>3.10E-5</td>
<td>Kedai Bunda</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>-7.023381</td>
<td>110.403030</td>
<td>-7.023381</td>
<td>110.403030</td>
<td>1.6E-6</td>
<td>2.5E-5</td>
<td>Unika Soegijap ranata</td>
</tr>
<tr>
<td>4</td>
<td>52</td>
<td>-7.062822</td>
<td>110.433580</td>
<td>-7.062746</td>
<td>110.433597</td>
<td>7.6E-6</td>
<td>1.69E-5</td>
<td>Koenang - Koenang</td>
</tr>
<tr>
<td>5</td>
<td>78</td>
<td>-7.058649</td>
<td>110.428642</td>
<td>-7.058651</td>
<td>110.428608</td>
<td>2.00E-6</td>
<td>3.39E-5</td>
<td>Setya Dimsum</td>
</tr>
</tbody>
</table>

Table 5.3: GPS Accurate

From the experimental results it can be seen that the Neo-7m Ublox GPS can provide a reasonably accurate location accuracy, but not 100% because it is slightly sliding from the actual location. To start up until satellite signals are obtained, when in the room takes longer time than outdoors.
4. ESP8266 send to thingspeak

In the performance test on ESP8266 is the process of sending data from arduino to thingspeak by using wifi Indihome.

<table>
<thead>
<tr>
<th>No</th>
<th>Wifi / Hotspot</th>
<th>Send to thingspeak (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Indihome</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>Indihome</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>Indihome</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>Indihome</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>Indihome</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5.4: WiFi ESP8266

From the experimental results, it can be seen that ESP8266 can send data to thingspeak by using wifi Indihome with a speed of approximately 20 seconds.

5. Wavecom SMS gateway gammu

In the performance test Wavecom will try using simcard Indosat IM3 and will try to send sms to five sim provider.

<table>
<thead>
<tr>
<th>No</th>
<th>Send SMS (second)</th>
<th>Provider (send to)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>XL</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>Mentari</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>Indosat IM3</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>Tri</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>Simpati</td>
</tr>
</tbody>
</table>

Table 5.5: SMS Gateway using Wavecom

From the experimental results, it can be seen that the gateway Sms gammu using Wavecom which contains simcard Indosat IM3 sends sms less than 20 seconds.
6. PHP calculate using Haversine Formula

In the performance test of PHP calculation using Haversine formula is to use the position of latitude and longitude in Graha Estetika Raya area with five different locations.

<table>
<thead>
<tr>
<th>No.</th>
<th>Latitude to</th>
<th>Longitude to</th>
<th>Latitude from</th>
<th>Longitude from</th>
<th>Distance (km)</th>
<th>Google Map (km)</th>
<th>Errors (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-7.061060</td>
<td>110.431113</td>
<td>-7.055172</td>
<td>110.429748</td>
<td>0.671</td>
<td>0.669</td>
<td>0.0021</td>
</tr>
<tr>
<td>2</td>
<td>-7.061060</td>
<td>110.431113</td>
<td>-7.023381</td>
<td>110.403030</td>
<td>5.21</td>
<td>5.21</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>-7.061060</td>
<td>110.431113</td>
<td>-7.054420</td>
<td>110.430541</td>
<td>0.741</td>
<td>0.740</td>
<td>0.0009</td>
</tr>
<tr>
<td>4</td>
<td>-7.061060</td>
<td>110.431113</td>
<td>-7.062822</td>
<td>110.433580</td>
<td>0.335</td>
<td>0.334</td>
<td>0.0001</td>
</tr>
<tr>
<td>5</td>
<td>-7.061060</td>
<td>110.431113</td>
<td>-7.058649</td>
<td>110.428642</td>
<td>0.382</td>
<td>0.382</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table 5.6: Haversine Formula Result

From the experimental results, it can be seen that the calculation with the Haversine Formula compared with the calculation of Google's distance measurement is almost 100% accurate. Since there are only 4 errors from 5 experiments below 1 km.

7. Fire Sensor KY-026 and PHP

In performance test Fire Sensor KY-026 to send SMS gateway gateway is to calculate how long the sensor so detect fire until haversine calculation then send sms.

<table>
<thead>
<tr>
<th>No.</th>
<th>Fire sensor detect fire (second)</th>
<th>PHP Calculate Haversine Formula (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6</td>
<td>0.277</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>0.342</td>
</tr>
<tr>
<td>3</td>
<td>0.8</td>
<td>0.471</td>
</tr>
<tr>
<td>4</td>
<td>0.8</td>
<td>0.711</td>
</tr>
<tr>
<td>5</td>
<td>0.6</td>
<td>0.641</td>
</tr>
</tbody>
</table>

Table 5.7: Performance Fire Sensor and PHP

From the experimental results, it can be seen that the detection of fire by the KY-026 sensor works under one second, while for the calculation of Haversine takes under one second.
5.2.2 Functional Test

1. Fire Sensor KY-026

The state of the flame sensor when there is no fire, visible led opposite the bright red led does not light. If the flame sensor is brought close to the flame, the red led indicator will illuminate both.

Illustration 5.21: Led Fire Sensor

Illustration 5.20: Led Fire Sensor (2)

2. ESP8266

When the ESP8266 is turned on, the red led on ESP8266 will always be on in the internet connected state or not.

If the blue led on ESP8266 blinks indicates that it is connecting to the internet. Once connected the blinking blue led light will die, but will blink again when doing data transmission.

3. GPS Ublox Neo-7M

The red led light on GPS Neo-7m indicates that the module is on, but still in the process of searching for satellite signals. The yellow led light on the bottom right will blink when the GPS has got a satellite signal and can transmit
latitude and longitude data of the current location. Here is a module image that is
in the room which takes about 2 minutes to get the satellite signal.

In the picture below is a picture module that is outside the room which takes about
1 minute to get the satellite signal.
4. **Wavecom**

Wavecom that has been connected to the computer will appear to work with a red blinking red mark.

Also when use gammu-identify, wavecom that indentified is ready to use.