CHAPTER 4
ANALYSIS AND DESIGN

4.1 Analysis

This research uses C language on Arduino for the encryption process and php for data decryption. The components needed in this research include:

1. 2 Arduino UNO
2. Ethernet Shield
3. LAN cable
4. Current sensor (ACS712 5V)
5. Adapter
6. Capacitor 100nf
7. Diode 1n4148
8. Jumper cable
9. Light sensor (LDR)
10. Potentiometer

There are 2 Arduinos which is the 1st Arduino is to execute DES/AES algorithm and connected with sensor and ethernet shield to send data, and the 2nd Arduino is to calculate the power consumption by 1st Arduino to execute the program.

4.1.1 DES and AES in Arduino

There are two cryptography algorithms in Arduino for the data encryption process, there are DES and AES algorithms. Data from the sensor is in the form of a string that will be encrypted and tested alternately using the DES and AES algorithms.

For DES, this project uses a library from the source https://github.com/Octoate/ArduinoDES/ . Because DES is only able to encrypt data by 8 bytes optimally, modifications are made so that even long data can be encrypted at once. The used method is to break the existing text/data into per 8 bytes. So the encryption

is done alternately per piece, then put back together into 1 string. But in the implementation using a sensor, the data from the sensor is set so the result becomes fix 8 bytes. The key in DES requires 8 bytes of data.

In AES, libraries are used from source https://www.arduinolab.net/aes-encryptiondecryption-using-arduino-uno/ . The AES plaintext can be optimally encrypted at a size of 16 bytes. A modification is made, so the encryption can be done with data multiples of 16 bytes. Because the size of the plaintext can only multiply by 8 bytes (in DES) and 16 bytes (in AES), then the text that is not even multiples of 8 or 16 bytes will be lost or not encrypted. For the AES key, 16 bytes of data are needed.

The ciphertext sent to the MySQL database server via an ethernet shield that has been pinned on Arduino. The picture can be seen on below:

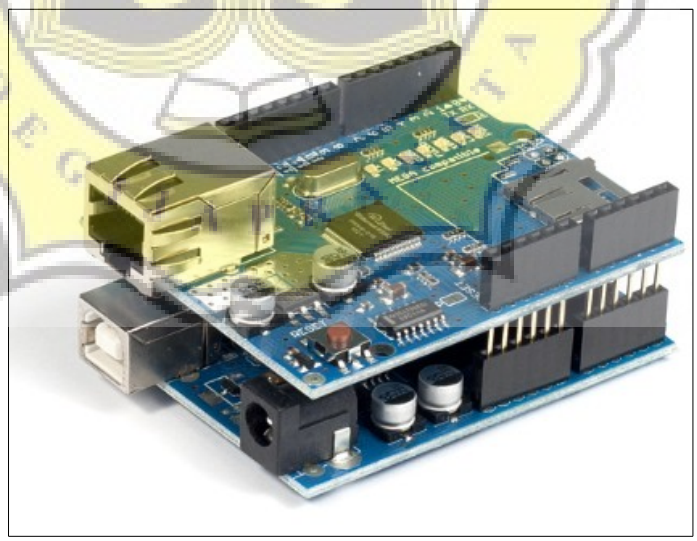

Illustration 4.1: Ethernet Shield pinned on 1st Arduino
(source : http://www.alselectro.com/ethernet-shield.html)

4.1.2 DES and AES in PHP

The ciphertext decryption process is done using php, the ciphertext data comes from Arduino sent via HTTP POST using an ethernet shield.

In DES decryption, this project uses a library from the source https://github.com/gilfether/phpcrypt. No need to separated the encryption data, all string data can be directly decrypted by php files. Enter the same key as used in encryption, because if the key is different then the decrypted result will not same with the plaintext.

In AES decryption, this project uses a library from the source https://github.com/phillipsdata/phpaes. Like DES, decryption of AES also doesn't need to separated per 16 bytes. The key is also entered with the same key as used in encryption.

4.1.3 Sensor

There are many kinds of sensor for telemedicine such as pulse sensors, body temperature, heart rate sensors, and etc. In this research, as a simulation, the sensor was replaced with LDR and potentiometer. LDR and potentiometer are connected with an ethernet shield that has been connected to 1st Arduino. The data from both sensor entered into the same coding as DES / AES. So the data that has been obtained can be directly encrypted and sent to the server.

4.1.4 Encryption Speed

The encryption speed measurement is done in the DES / AES program using micros() function. There are two calculation of time used for the encryption process. The first time is encryption per piece. Second, the encryption time of all strings/data. The result of micros() is a time in microseconds, note that 1 micros = 0.000001 seconds.

4.1.5 The Power Consumption

The power consumption by the 1st Arduino is measured by ACS712 sensor that connected with 2nd Arduino. This research used the ACS712 sensor with the addition of diode 1n4148 and 100nf capacitors. The use of diodes is to control the direction of the current so that the current through the diode is only one direction. While the use of capacitors is to filtering the ripples from the rectification so we obtained the smooth and flat waves. Both of these components are intended so that the results of the ACS712 sensor can be more stable.

For the scheme of the ACS712 sensor can be seen as follows.

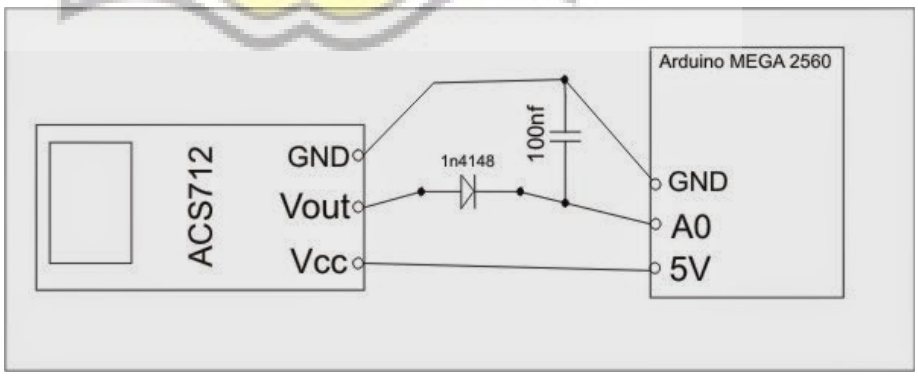

Illustration 4.2 : Power Measurement Scheme
(source : http://jawaplugin.blogspot.com/2014/08/mengatasi-sensor-arus-acs712-tidak.html)

ACS712 is connected to an adapter and 1st Arduino DES / AES. ACS712 pins are connected like in the scheme, GND is connected with capacitors 100nf and GND on Arduino to measure current. Vout is connected with diodes and capacitors and 2nd Arduino analog pins (A0). Vcc is connected with a 5V pin on 2nd Arduino.

4.2 Desain

4.2.1 Data Encryption

Here is the overall Flowchart of DES and AES data encryption processes with sensors on 1st Arduino.

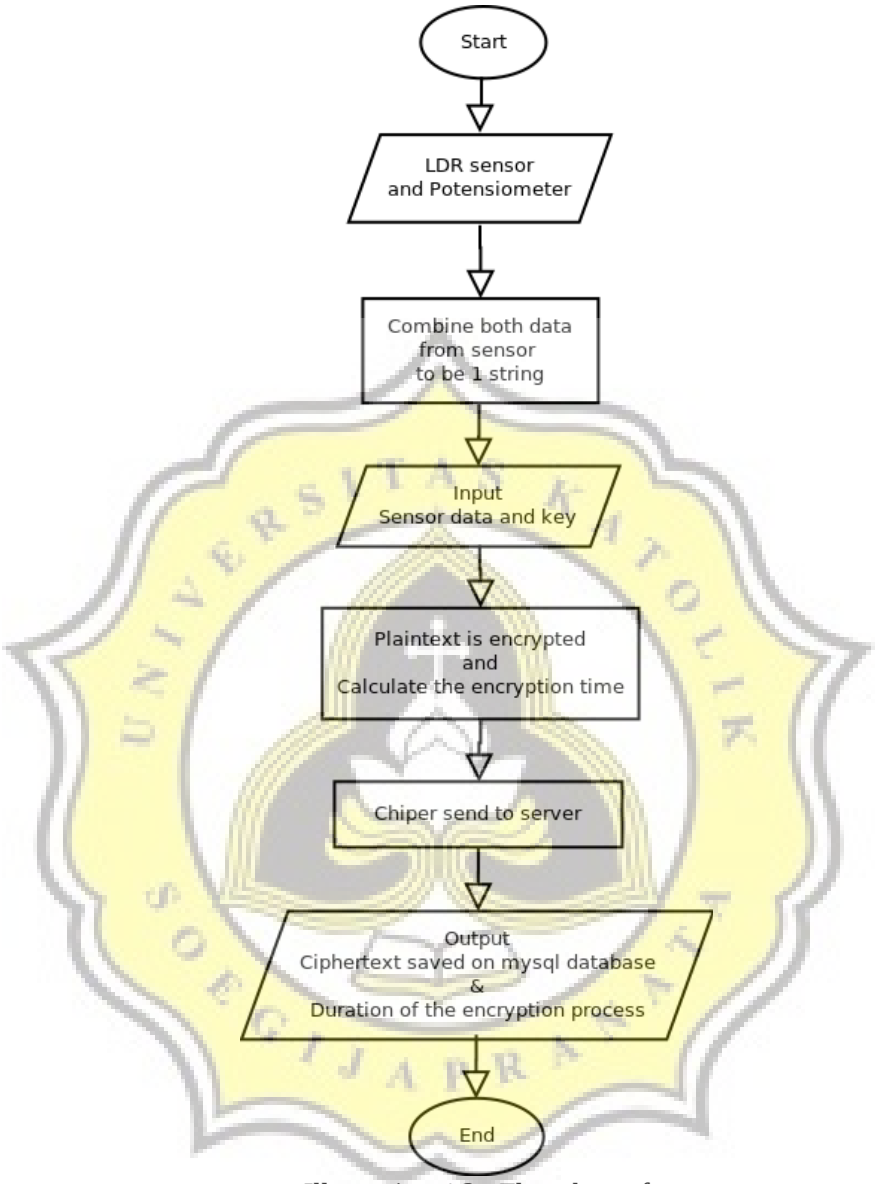

Illustration 4.3 : Flowchart of
Encryption Process

In data encryption, the first thing to do is read data from the LDR sensor and the Potentiometer. The results of the two sensors are combined into a string with a length of 16 bytes. Enter 8 digit key (8bytes) for DES, and 16 digit key (16 bytes) for AES. When the encryption process takes place there are two calculation of processing time, there are time per 8 bytes of data (in DES) and 16 bytes of data (in AES) and the entire time of the encryption

process. The time used for the encryption process can be seen in the serial monitor as a comparison material. The program also produces ciphertext which is sent directly to the server via HTTP POST.

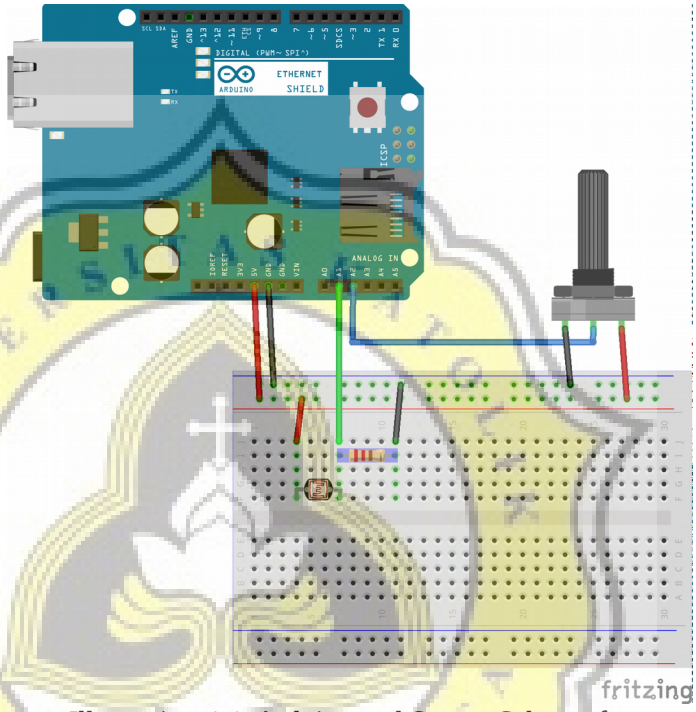

Illustration 4.4: Arduino and Sensor Scheme for Encryption

4.2.2 Data Decryption

Here is the flowchart of the data decryption process in both DES and AES in PHP.

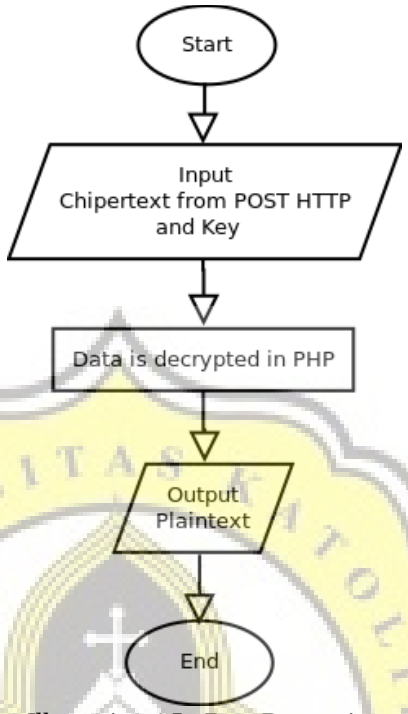

Illustration 4.5 : Data Decryption Flowchart

Data from Arduino sent via HTTP POST will be processed by php. Ciphertext will be decrypted using the same key as the encryption process. The result of the decryption is the plaintext from sensor data result sent by 1st Arduino.

The encryption process is done on Arduino because most remote inspection devices, mainly for retrieving patient data are using sensors connected to Arduino. The encryption process is carried out on the same program when reading the results of sensor data. So the data is directly processed and not read or changed by unauthorized parties. The process also requires a lightweight program because there is a possibility that the tools used for data retrieval require little power, such as from a battery.

The decryption process is done using php because the data that has been sent to Arduino via POST HTTP will be received by the user even from different city. In order to read directly, the

decryption process is done using php without having to reconnect to Arduino at the destination.

4.2.3 Power Consumption

Here is the flowchart of power consumption by the 1st Arduino where the DES/AES encryption process occurs.

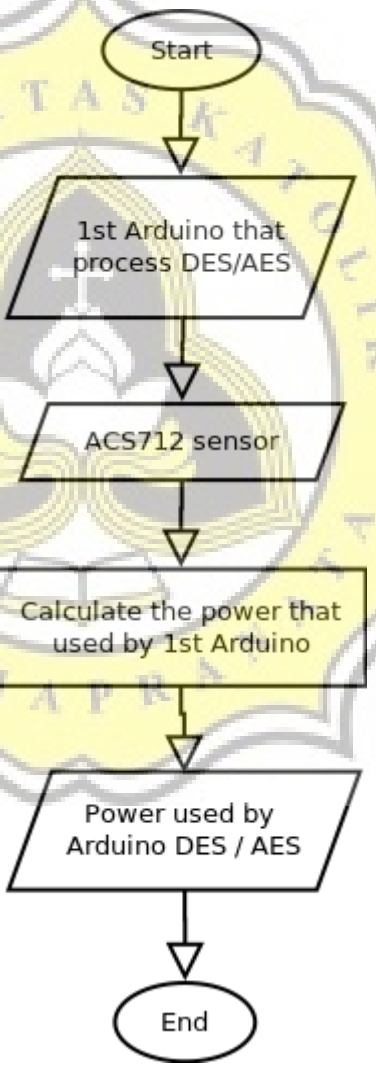

Illustration 4.6: Flowchart of Power Consumption by Arduino

ACS712 sensor that connected to 2nd Arduino will do the power reading that is used by 1st Arduino (DES/AES process). The results of reading the ACS712 sensor are displayed in the serial monitor as a comparison material for which algorithms use more power.

4.2.4 Complete Circuit

Below is the circuit scheme made in this research.

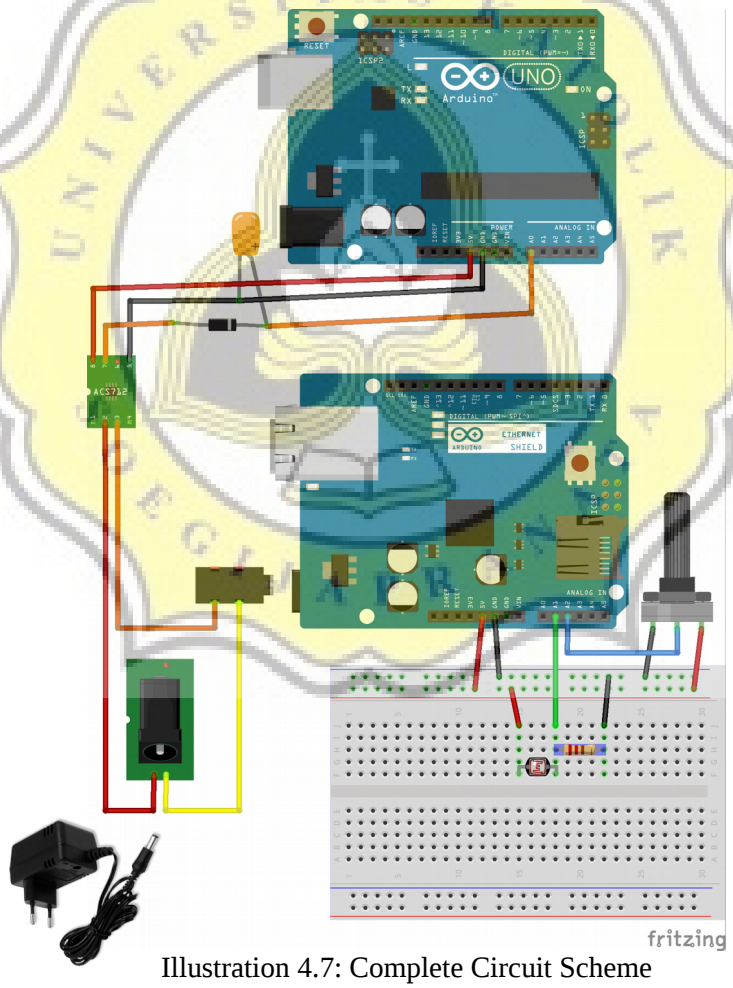

Illustration 4.7: Complete Circuit Scheme

After the data from the LDR and Potentiometer sensors are read, the data will be processed in the 1st Arduino and the encryption process will be carried out. The power obtained by the 1st Arduino comes from the jack that is connected to the adapter. Then the 1st Arduino is connected to the ACS712 sensor to do the power reading used on 1st Arduino. Furthermore, the ACS712 sensor is connected to the 2nd Arduino by passing diodes and capacitors so the results of the obtained power measurements are more stable and accurate. The power measurement results appear in the serial monitor for a comparison between the power output used by DES and AES.