The usage of Information and Technology in Indonesia is increasing from year to year. Internet is growing rapidly and prices offered by Internet providers are getting cheaper over time. In addition, smartphone users in Indonesia is also growing in number. This leads to the dependence between the user technologies with information technology systems. In the emerging era, a lot of activities that can be completed by the information system. This makes the users of such technology was spoiled by the presence of new technologies. This prompted the authors to research and design a home automation system that can be used by the user to adjust the electricity consumption by appliances that use electricity at home by using android application that is embedded in the mobile device so that the electricity consumption in the home is not excessive and needed when necessary. Design of the home automation system designed from data collection to conduct interviews regarding the structure of electricity at home, making home electrical schematic design, code generation commands to the microcontroller, the design of the web server for communication with mobile application to microcontroller, the design of the server for home automation. It is used to schedule each electrical appliance that is registered with the manufacture of a mobile application for home automation system that will be embedded in mobile devices and the Android Operating System.

**Keywords:** mobile, house automation, electricity

1. Chapter I Introduction

1.1 Background Analysis

Excessive power consumption is not efficient given The electricity prices in Indonesia has a significant increase and energy waste also less efficient for the state of our Earth remembering that a lot of people who need that energy. In this growing era, many scientists are trying to research the absence of energy waste by researching to improve the energy substitution in the Earth’s energy cycle.

The Role of Information Systems in this case to be one solution. For example, an appropriate information system can be useful to help users organize the use of peripheral devices that are within range of the user’s home. With this information system, hopefully the use of electricity at home user is more efficient.
1.2 Problem Analysis

1. How to make House Automation System?
2. How to communicate Micro Controller to Device and to Server?
3. How to make Android Mobile Application that can communicate with Micro Controller?
4. How to make Scheduling system that reorganize every devices that already signed up?

1.3 Research Scope

Scope of this research are:

1. Using Android 2.3.0 or Higher.
2. Using PHP Based Web Server.
3. The Specification of PC that was used:
   - Pentium 4 or higher
   - 1GB RAM or higher
   - 128GB hard disk or higher
   - Has USB Port
   - Has Ethernet Port.
4. Security issues are not included.
5. Using 3G Network.
8. The devices used for trial are LEDs.

1.4 Research Purpose

Making a house automation system with Android Application as the control panel of the device listed.

1.5 Research Benefit

House automation systems facilitate the user to set any appliance without having to physically control it through the switch.

1.6 Entrepreneurial Aspects

The value of Opportunity that contained is Mobile Home Automation System can be adjusted by the user’s will.

The value of Market Sensitivity, Indonesian technological developments are beginning to switch from manually towards automation.

Creative and Innovative values described in the phrase "home automation with remote control", which organize home appliances, and can be accessed anywhere and anytime.
1.7 Research Methodology

- Interview / Questionaire
- Literature Study Electricity Scheme
- Study C++ Language
- Study PHP Language
- Developing Java Application
- Developing Android Application
- Developing House Prototype
- Implementing System
- Making Conclusion
- Making Report

2. Chapter II Theory

2.1 Arduino

One brand of Microcontroller that has Open Source programming language. ([ref:arduino.cc](http://arduino.cc))

2.2 Android Application

2.2.1 Activity

There are 6 Android’s Activity Screens:

2.1.1 Log In Activity

Works for validating the users.

2.2.1.2 Home Activity

Navigator to another Activity.
2.2.1.3 Status Screen Activity

To view the status of the device listed in the House Automation System’s user.

2.2.1.4 Setting Activity

To view the schedule of each device listed in the House Automation System.

2.2.1.5 Info Activity

Intended as a display electric’s fee information of each column.

2.2.1.6 IP Setting Activity

Set House Automation System’s IP Address.

2.2.2 Session

The Method made to fulfill the functions of a session at the turn of Android that will be used is the method SharedPreferences. Examples of the code of SharedPreferences:

```java
SharedPreferences sp = getSharedPreferences("loginPref", Context.MODE_PRIVATE);
Editor edit = sp.edit();
edit.clear();
edit.putBoolean("succeed", true);
edit.commit();
```

The goal is when Loged In, this code is called, and Shared Preferences loginPref succeed filled with Boolean parameter to true.

2.2.3 Layout

Layout used for Activity are Vertical and Horizontal LinearLayout.

![Figure 2.2.3 Layout Android](image)

2.2.4 Intent

Navigates one page to another page. One of its uses:
Intent i = new Intent(getApplicationContext(), Home.class);
startActivity(i);

2.2.5 HTTP POST

The method used by Android Core to interact with PHP files with extensions.

2.3 Database

2.3.1 MySQL

SQL data base management systems.
(ref: http://www.mysql.com/why-mysql/)

3. Chapter III : System Analyze and Design

3.1 Preliminary Research

Here are the results of the preliminary research by 100 samples:

Lamp is dominant electric device.
Saving electricity is important.
Electric device schedule is important.
Electric devices dominant turned on 14.00-22.00 WIB.
Electricity’s fee is important.
UI dominant on blue print.

System Architecture

Figure 3.2 System Architecture
3.3 Interface Flow Diagram

3.3.1 Android Application

![Android Application Interfaces Flow](image1)

Figure 3.3.1 Android Application Interfaces Flow

3.3.2 Java Desktop Application

![Java Application Interfaces Flow 1](image2)

Figure 3.3.2.1 Java Application Interfaces Flow 1

![Java Application Interfaces Flow 2](image3)

Figure 3.3.2 Java Application Interfaces Flow 2

3.4 ER Diagram

![ER Diagram](image4)

Figure 3.4 ER Diagram
3.5 Communication Between Micro controller, PC, and Device.

3.5.1 Serial Connection Trial

Arduino driver for PC downloaded via www.arduino.cc. Parameters needed on the Arduino IDE are COM port the Arduino plugged in and the type of the board is Uno.

![Figure 3.5.1.3 Setup Parameter for Arduino Uno.](image)

Then open Blink.ino sample code, and upload the code to the Arduino Uno that is still plugged in the USB Serial. Then LED beside the pin 13 will blink.

![Figure 3.5.1.4 Uploading Blink ke Arduino Uno](image)

3.5.2 Direct Connection from Arduino Uno into device

Take 2 jumper cables with black and red Shelled its end, and an LED. Connect the black jumper cable to ground pin and the red jumper cable to the pin number 13 on the Arduino Uno board. Long LED foot connected to red cable, and short LED foot connected to black cable. If successful, the LED will blinking.

3.5.3 Serial Communication from PC to device

Here is the example code on Arduino Uno :

```cpp
void setup() {
  Serial.begin(9600);  // initialize serial communication:
  pinMode(13, OUTPUT);  // initialize output
}
void loop() {
  // read the serial port:
```
if (Serial.available() > 0) {
    int inByte = Serial.read();
    Serial.println(inByte);
    switch (inByte) {
        case 'u':
            digitalWrite(13, HIGH);// set the LED on
            break;
        case 'v':
            digitalWrite(13, LOW);// set the LED off
            break;
    }
}

The purpose of the above code, a message can be sent to Arduino Uno board through char letter is entered in Hyper Terminal application. If the message sent is 'u' then the pin number 13 will live and vice versa.

3.6 Android Application Communicate with Microcontroller

Make the PHP configuration file to perform serial communication to microcontroller through USB Serial when the PHP file is accessed through Android Application. Emphasized to set router’s DMZ setting so the data will be transferred to the PC’s ip address which has the PHP files.

3.6.1 PHP Turn On/Off Device

The key to communicate with microcontroller. Example code-named checkpin.php:

```php
…
exec('mode COM7: baud=9600 data=8 stop=1 parity=n xon=no');
// COM7 = Location of Arduino Uno PORT.(no 7 pointed the location of the PORT)
// baud = bit rate per second, by default 9600
// data = data bits, by default = 8
// stop = stop bits, by default = 1
// parity = filled by n that means 'none'(default)
// xon = no by default
include('koneksi.php');
$pin = $_POST['pin'];
$switch = $_POST['status'];
switch($pin){
    case "1":
        switch ($switch) {
            case "on":
                // serial communication for port COM7
                $fp = fopen("COM7:", "w+"));
                if (!$fp) {
                    echo "Port cant be Opened.";
                }
```
The purpose of the code above is to turn on the device on pin 3 by accessing the php file above which functions to provide input in the form of char 'a' that it is ASCII-97 and vice versa with char 'b'.

### 3.6.2 Android Activity Post

PHP function to access files created in the previous section by using HTTP POST. The following example HTTP POST on one Android Activity:

```java
public void postPHP(){
try{
    final String statusStr = "on";
    client =new DefaultHttpClient();
    post = new HttpPost("http://"+GlobalVar.ip+"/smarthome/checkpin.php");
    namevaluepair = new ArrayList<NameValuePair>();
    namevaluepair.add(new BasicNameValuePair("pin", pinStr));
    namevaluepair.add(new BasicNameValuePair("status", statusStr));
    post.setEntity(new UrlEncodedFormEntity(namevaluepair));
    ResponseHandler<String> responseHandler = new BasicResponseHandler();
    final String responseStr = client.execute(post, responseHandler);
    client.getConnectionManager().shutdown();
    runOnUiThread(new Runnable() {
        @Override
        public void run()
        {
            //do something here
        }
    });
}
```
3.7 Scheduling System on House Automation System

3.7.1 PHP Turn On/Off Alat Tertentu

To facilitate the scheduling system of Home Automation System designed, make some PHP files that have a file name based on the different functions. Here's an example of a php file named onpin1.php:

```php
... 
exec('mode COM7: baud=9600 data=8 stop=1 parity=n xon=no');
// COM7 = Location of Arduino Uno PORT. (no 7 pointed the location of the PORT)
// baud = bit rate per second, by default 9600
// data = data bits, by default = 8
// stop = stop bits, by default = 1
// parity = filled by n that means ‘none’ (default)
// xon = no by default
$fp = fopen("COM7:", "w+");
// serial communication for port COM7
if (!$fp) {
  echo "Port cant be Opened.";
} else{
  fwrite($fp, chr(97)); // write ‘a’
  fclose($fp); // close serial communication
}
... 
```

The purpose of making the above php file, when the file is accessed by desktop application later, the LED on pin number 3 (first pin is used) will live.

3.7.2 Timer Java Application

Aims to check the time to turn off / turn on the device on the Arduino Uno board via serial Communication by accessing the PHP files. Example of the Code:

```java
public void setTime()
{
timer = new javax.swing.Timer(1000, new ClockTickAction());
timer.start(); // start timer
}
private class ClockTickAction implements ActionListener{
    public void actionPerformed(ActionEvent e){
        Calendar now = Calendar.getInstance();
        hour = now.get(Calendar.HOUR_OF_DAY);
    }
```
minute = now.get(Calendar.MINUTE);
second = now.get(Calendar.SECOND);
sysTime = new String (hour + ":" + minute+ ":" +second);
checkSchedule(); //check schedule
}
}

ClockTickAction timer called by the setTime() method when the application was run for the first time. Then for checkSchedule method has code like this:
public void checkSchedule(){
int id=1;
for(int i=0; i<counter-1; i++){*
    scheduleVec = null;
    // vector scheduleVec inputed from database
    scheduleVec = new ScheduleJDBCDAO().read("select * from schedule where devices_pin ='"+id);
    // inputing ScheduleDom with vector scheduleVec
    ScheduleDom sd = (ScheduleDom)scheduleVec.elementAt(0);
    //comparing time on/off with current time
    if(java.sql.Time.valueOf(sysTime).equals(sd.getTimeOn())){
        turnOn(id);
    }else if( java.sql.Time.valueOf(sysTime).equals(sd.getTimeOff())){
        turnOff(id);
    }
    id++;
}
}

The purpose of writing the above code is doing the examination schedule in the database timeOff and timeOn for every devices. Then the structure of the code method turnOn(id) or turnoff(id) is:
public void turnOn(int id){
DevicesJDBCDAO d = new DevicesJDBCDAO();
deviceVec = null;
//inputing vector deviceVec from database
deviceVec = d.read("select * from devices where pin='"+id);
//inputing domain DeviceDom with vector deviceVec
DevicesDom dd = (DevicesDom)deviceVec.elementAt(0);
int status = dd.getDeviceStatus();
//ambil nilai status dari dd
if(status==0){
    try{
        //open file PHP that has serial communication function
        URL url = new URL( "http://localhost/smarthome/onpin"+id+".php" );
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        if( conn.getResponseCode() == HttpURLConnection.HTTP_OK ){
            InputStream is = conn.getInputStream(); // open file PHP
        }else{
            InputStream err = conn.getErrorStream();
        }
    }catch (Exception e){
        e.printStackTrace();
    }
}
The purpose of writing the above code is to turn on the devices on parameter id that thrown by checkSchedule() method.

### 3.7.3 Scheduling on Android Application

Activity for scheduling also has the HTTP POST method. Here's an example of using the HTTP POST:

```java
try {
    String devices_pin = "";
    String time_on = "";
    String time_off = "";
    for (int i = 0; i < datasize; i++){
        devices_pin = devices_pin + pin[i] + "-";
        time_on = time_on + timeOn.get(i).getText().toString() + ":00" + "-";
        time_off = time_off + timeOff.get(i).getText().toString() + ":00" + "-";
    }
    client = new DefaultHttpClient();
    client.getConnectionManager().closeExpiredConnections();
    post = new HttpPost("http://" + GlobalVar.ip + "/smarthome/updateschedule.php");
    namevaluepair = new ArrayList<NameValuePair>();
    namevaluepair.add(new BasicNameValuePair("devices_pin", devices_pin));
    namevaluepair.add(new BasicNameValuePair("time_on", time_on));
    namevaluepair.add(new BasicNameValuePair("time_off", time_off));
    post.setEntity(new UrlEncodedFormEntity(namevaluepair));
    ResponseHandler<String> responseHandler = new BasicResponseHandler();
    final String responseStr = client.execute(post, responseHandler);
    client.getConnectionManager().shutdown();
    //do something else maybe
} catch(Exception e) {
}
```

The purpose of the above code is when it was accessed by one entity Android Activity, the Activity Android will access the PHP files in the code to throw the appropriate parameters.
4. Chapter IV: Implementation and Testing

4.1 Scenario Testing

4.1.1 Log In to Server

1. The user enters a username and password that have been made to the menu Log In.
2. When finished entering data, the user presses the Sign In button.

4.1.2 Log In To Mobile Application

1. Users open a Mobile Application in Android named autogen.
2. The user enters a username, password, and IP address on the menu Log In.
3. After inputting all the values, the user presses the OK button.

4.1.3 View Device's Status and Turn On / Off the device

1. Once inside the Home Menu autogen, user pressing the existing status.
2. Users turn on one device by pressing the 'OFF'.

4.1.4 Setup Schedule for devices

1. User on the Home Menu Mobile Application presses Settings button.
2. Furthermore, the user presses the Edit button at bottom of the user’s screen.
3. The user presses a button, and a dialog timeOn and time settings will pop up.
4. Users enter the desired time, and press the OK button on the dialog.
5. The user presses the Save button on the Settings screen.

4.1.5 View Device's Electric Fees

1. User on the Home Menu application AutoGen pressing the Info button.
2. On the next page given the information of electricity costs of the device.

4.1.6 Log Out from Mobile Application

1. User on the Home Menu AutoGen pressing the Exit button.
2. Confirmation dialog will appear and the user presses the Yes button.

4.2 Results of Testing

For results based on testing scenarios conducted almost did not have any error such as the following:
1. Character input is not appropriate
2. Notifications do not appear
3. Error within session displacement
Mistakes outside the tested scenarios can be happened, but tested scenarios were optimal.
5. **Chapter V : Conclusion and Advice**

5.1 **Conclusion**

The Conclusion Based on the analysis and design to implementation of the system are:

1. House Automation is designed to answer the Problem Analysis.
3. Implementation System for Home Users resulting reshuffling the electrical wiring.
4. Android device with screen higher than 3.5” given applications with different layout arrangement.
5. Wearing Gmail as Mail Server because Gmail can display HTML emails.
6. User Interface using table view because it was about to be mass-produced. House automation system is designed according to the respondent wishes.

5.2 **Advice and Development**

Berikut saran yang diberikan oleh penulis:

1. Implementation System in conjunction with the CCTV system is highly recommended.
2. Mobile Application User Interface can be requested accordance the house’s blue print.
3. Systems can be added with further recommendations.
6. References


