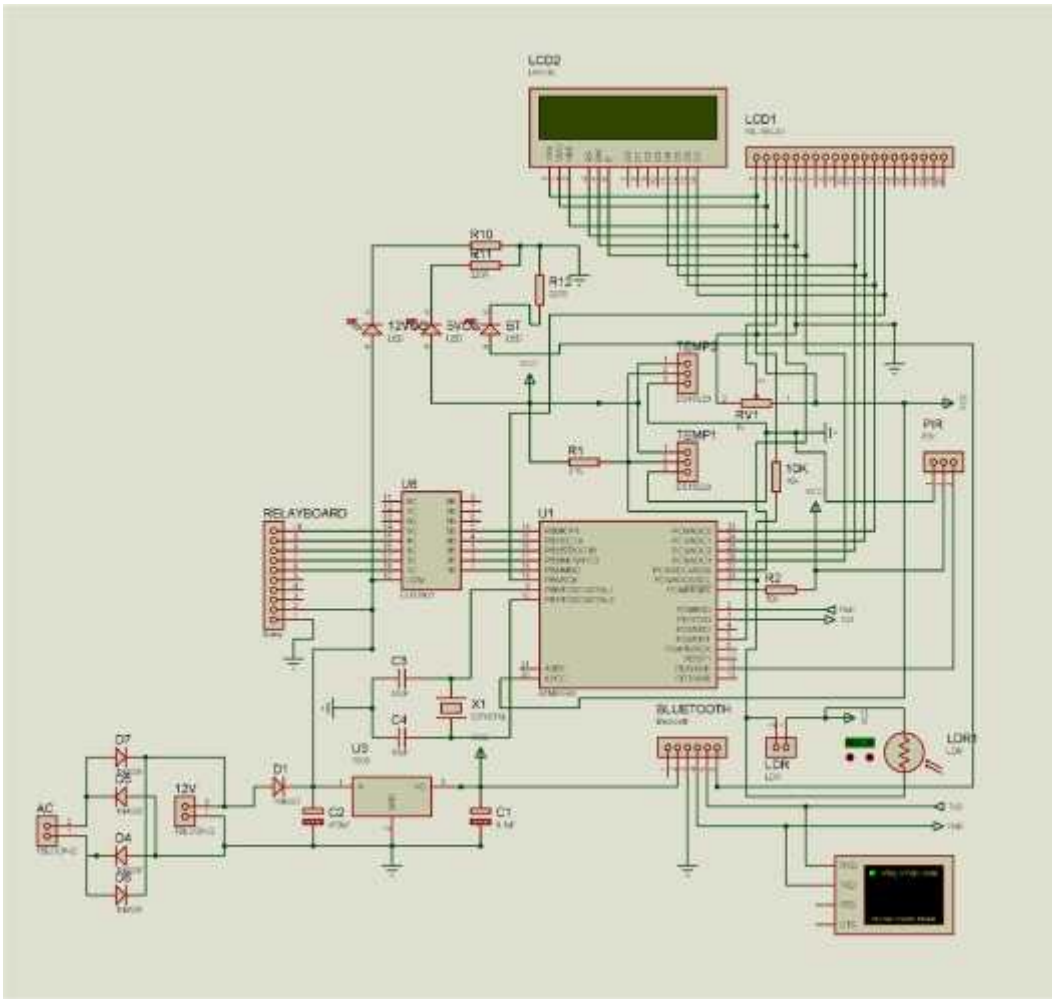
BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Perangkat Keras

Berikut adalah beberapa komponen perangkat keras dari Sistem Rumah Pintar Sederhana:

4.1.1 Mikrokontroler AVR ATMega8



Gambar 4.1 Skema Mainboard

Mikrokontroler Atmel ATMega8 adalah bagian inti dari seluruh bagian interface Rumah Pintar Sederhana, sebab bagian ini terdapat mikrokontroler Atmel ATMega8 yang isinya adalah software Rumah Pintar Sederhana, mikrokontroler ATMega8 membutuhkan tegangan 5V dan Ground, untuk dapat berfungsi dengan baik mikrokontroler ATMega8 diberikan sistem detak Clock dengan menggunakan komponen kristal internal sebesar 1 MHz.

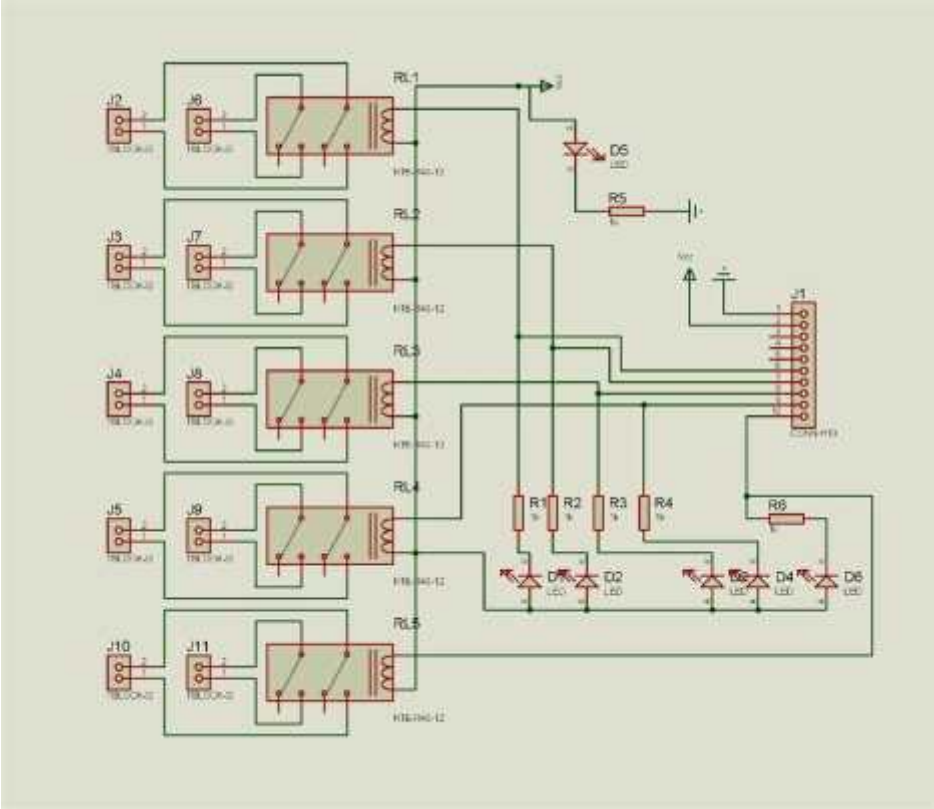
4.1.2 Power Supply

Sistem Rumah Pintar Sederhana tidak akan berfungsi dengan baik tanpa mendapat tegangan dari power supply. Power supply berisi komponen-komponen yang bertugas mengubah Arus AC menjadi Arus DC, kemudian Arus DC tersebut distabilkan menurut kebutuhan Mikrokontroler dan sistem Mikrokontroler, pembuatan Power Supply hasil keluaran out put harus benar-benar teregulasi, karena dalam pemakaiannya berhubungan dengan sebuah komponen yang peka terhadap suatu batas tegangan. Untuk mendapatkan hasil yang sempurna sesuai tegangan masukkan dan keluaran, maka dari itu perlu diperhitungkan dengan seksama agar nanti dalam pemakaian tidak terjadi trouble atau kesalahan yang tidak diinginkan, kemungkinan akan drop tegangannya atau tidak stabil outputnya bila setelah terbebani.

4.1.3 Sistem Relayboard

Cara kerja sistem ini adalah apabila mikrokontroler menerima perintah untuk mengubah data dari LOW menjadi HIGH atau sebaliknya yang kemudian akan dikirimkan ke amplifier ULN2803 dan diteruskan ke relayboard yang akan

menghidupkan relay dimana relay tersebut menjadi penghubung antara AC supply dan perangkat elektrik



Gambar 4.2 Skema Relayboard

4.2 Komunikasi Data

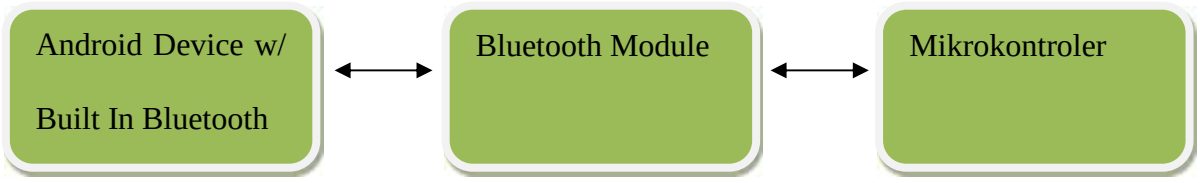
Berikut adalah gambar yang menjelaskan fungsi dari pinout Bluetooth HC-05 yang penulis gunakan untuk menjadi penghubung atau media komunikasi antara mikrokontroler AVR ATmega8 dengan *smartphone* android :



Gambar 4.3 Pinout dari HC-05

RXD merupakan pin untuk menerima data dari mikrokontroler AVR ATmega8, TXD merupakan pin untuk mengirimkan data dari mikrokontroler AVR ATmega8, sedangkan GND dan VCC merupakan pin untuk arus listrik 5V yang dikirimkan oleh *Power Supply*.

Gambar dibawah ini menjelaskan stuktur komunikasi data yang digunakan dalam Sistem Rumah Pintar Sederhana yang penulis buat:



Gambar 4.4 Diagram Sistem Komunikasi Data Bluetooth dengan Mikrokontroler

4.3 Perangkat Lunak

Dalam penelitian ini penulis menggunakan bahasa pemrograman Basic Compiler dan Java (Android Development Tools). Berikut adalah beberapa fungsi-fungsi yang digunakan:

4.3.1 Basic Compiler (BASCOM)

Program mikrokontroler didalamnya terdapat beberapa sub bagian program yang saling terkait, diantaranya :

4.3.1.1 Device Control

Sub program ini dijalankan untuk membaca data input yang dikirim oleh aplikasi android melalui Bluetooth untuk merubah status Switch 1-5.

```
Sub Device_control
  Tmp = Inkey()

  If Tmp = 65 Then                    'Dev 1 On A
    Set Devstat.4
    Portb = Devstat
  Elseif Tmp = 97 Then                'Dev 1 Off  a
    Reset Devstat.4
    Portb = Devstat
  Elseif Tmp = 66 Then                'B
    Set Devstat.3
    Portb = Devstat
  Elseif Tmp = 98 Then                'b
    Reset Devstat.3
    Portb = Devstat
  Elseif Tmp = 67 Then                'C
```

```
    Set Devstat.2
    Portb = Devstat
  Elseif Tmp = 99 Then                    'c
    Reset Devstat.2
    Portb = Devstat
  Elseif Tmp = 68 Then                    'D
    Set Devstat.1
    Portb = Devstat
  Elseif Tmp = 100 Then                    'd
    Reset Devstat.1
    Portb = Devstat
  Elseif Tmp = 69 Then                    'E
    Set Devstat.0
    Portb = Devstat
  Elseif Tmp = 101 Then                    'e
    Reset Devstat.0
    Portb = Devstat
  Elseif Tmp = 72 Then                    'ALL ON H
    Devstat = &HFF
    Portb = Devstat
  Elseif Tmp = 104 Then                    'ALL off h
    Devstat = &H0
    Portb = Devstat

  End If
            End Sub
```

4.3.1.2 Convallt

Sub program ini digunakan untuk mengetahui pada berapa bit serta membaca data pada sensor suhu *Dallas DS18B20*.

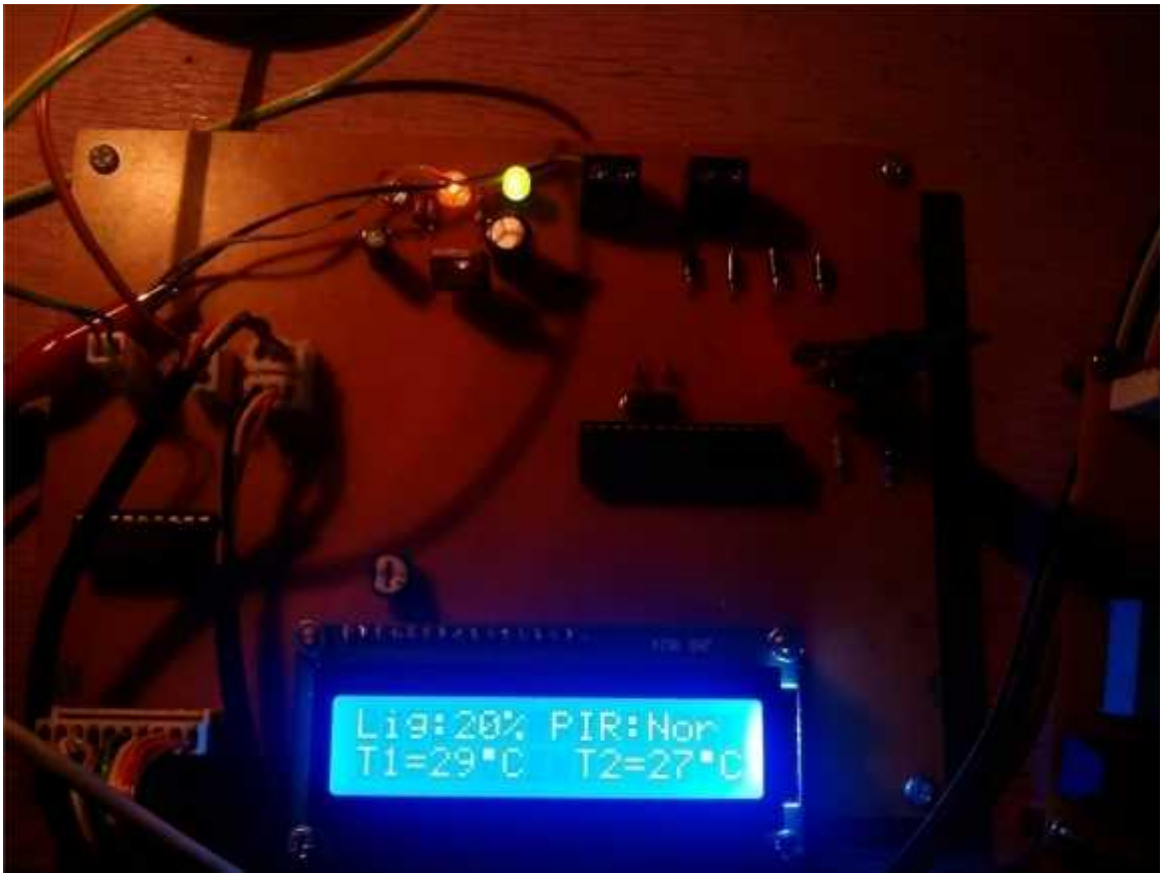
```
Sub Convallt
1wreset                          ' reset the bus
1wwrite &HCC                     ' skip rom
 1wwrite &H44                    ' Convert T
End Sub
```

4.3.1.3 Decigrades

Function ini digunakan untuk mengubah data digital yang dikirim oleh sensor suhu dan merubahnya menjadi satuan celcius dengan menggunakan rumus yang telah ditentukan oleh *datasheet* yang diberikan oleh *Dallas DS18B20*.

```
Function Decigrades(byval Sc(9) As Byte)
 Decigrades = 0
 Decigrades = Makeint(sc(1) , Sc(2))
 Decigrades = Decigrades / 16
End Function
```

4.3.2 Tampilan LCD pada mikrokontroler



Gambar 4.5 Tampilan pada LCD mikrokontroler

Pada LCD mikrokontroler akan terlihat nilai dari sensor suhu 1, sensor suhu 2, internsitas cahaya dari LDR, dan status gerakan dari sensor PIR.

4.3.3 Android Dev Tools (Java Eclipse)

Pada aplikasi android dibagi menjadi beberapa sub program,diantaranya adalah sebagai berikut:

4.3.3.1 Mencari perangkat mikrokontroler

Berikut adalah barisan coding untuk fungsi pencarian perangkat mikrokontroler:

```
void findBT() {
        try {
        mBluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();

            if (mBluetoothAdapter == null) {
            status.setText("No bluetooth adapter available");
            }

            if (!mBluetoothAdapter.isEnabled()) {
                Intent enableBluetooth = new Intent(

    BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBluetooth, 0);
            }
        Set<BluetoothDevice> pairedDevices =
mBluetoothAdapter
```

```
                .getBondedDevices();
                if (pairedDevices.size() > 0) {
                        for (BluetoothDevice device :
pairedDevices) {

                        if
(device.getName().equals("SmartHome")) {
                                mmDevice = device;
                                break;
                                }

                        }
                }
        //Toast.makeText(getApplicationContext(), " HAS Bluetooth
Device NOT Found", Toast.LENGTH_SHORT).show();
        //status.setText("Bluetooth Device Found");
                } catch (NullPointerException e) {
                        e.printStackTrace();
                } catch (Exception e) {
                        e.printStackTrace();
                }}
```

4.3.3.2 membuka koneksi Bluetooth dengan perangkat mikrokontroler

Berikut adalah barisan bahasa pemrograman menggunakan Android Development Tools untuk menghubungkan atau mengsinkronisasi dengan perangkat mikrokontroler :

```
void openBT() throws IOException {
    try {
    // Standard SerialPortService ID
    UUID uuid = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");
mmSocket = mmDevice.createRfcommSocketToServiceRecord(uuid);
            mmSocket.connect();
            mmOutputStream = mmSocket.getOutputStream();
            mmInputStream = mmSocket.getInputStream();

            beginListenForData();

            toggle1.setEnabled(true);
            toggle2.setEnabled(true);
            toggle3.setEnabled(true);
            toggle4.setEnabled(true);
            toggle5.setEnabled(true);
            toggleall.setEnabled(true);
```

```
                closeButton.setEnabled(true);
                openButton.setEnabled(false);
                exitButton.setEnabled(false);
        Toast.makeText(getApplicationContext(), "HAS Bluetooth Connected",
        Toast.LENGTH_SHORT).show();
            status.setText("HAS Bluetooth Connected");
            } catch (NullPointerException e) {
            e.printStackTrace();
            } catch (Exception e) {
            e.printStackTrace();
            }
        }
```

4.3.3.3 membaca data dari Bluetooth

Berikut adalah barisan bahasa pemrograman menggunakan Android Development Tools untuk fungsi membaca data yang dikirim oleh perangkat mikrokontroler:

```
        void beginListenForData() {
            try {
                final Handler handler = new Handler();

                // This is the ASCII code for a newline character
                final byte delimiter = 10;
```

```
stopWorker = false;
readBufferPosition = 0;
readBuffer = new byte[1024];

workerThread = new Thread(new Runnable() {
public void run() {

while (!Thread.currentThread().isInterrupted()
&& !stopWorker) {

try {

int bytesAvailable = mmInputStream.available();
if (bytesAvailable > 0) {
byte[] packetBytes = new byte[bytesAvailable];
mmInputStream.read(packetBytes);
for (int i = 0; i < bytesAvailable; i++) {
byte b = packetBytes[i];
if (b == delimiter) {

byte[] encodedBytes = new byte[readBufferPosition];
System.arraycopy(readBuffer, 0,
```

```
                    encodedBytes, 0,
                    encodedBytes.length);
                    final String data = new String(
                    encodedBytes, "US-ASCII");
                    readBufferPosition = 0;
                    handler.post(new Runnable() {
                    public void run() {
                    receive.setText(data); }
                    });
} else {
                    readBuffer[readBufferPosition++] = b;
                              }
                    }
                    }
                                        } catch (IOException ex) {
                                                  stopWorker = true;
                                        }}
                                        });
                    workerThread.start();
               } catch (NullPointerException e) {
                    e.printStackTrace();
               } catch (Exception e) {
                    e.printStackTrace();
```

```
        }
    }
```

4.3.3.4 Mengirimkan data ke perangkat mikrokontroler

Berikut adalah barisan bahasa pemrograman menggunakan Android Development Tools untuk fungsi mengirimkan data mikrokontroler :

```
  void sendData() throws IOException {
        try {

            // the text typed by the user
            String msg = myTextbox.getText().toString();
            msg += "\n";

  mmOutputStream.write(msg.getBytes());

            // tell the user data were sent
            status.setText("Data Sent");
                } catch (NullPointerException e) {
            e.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

4.3.3.5 Menutup koneksi Bluetooth dengan perangkat mikrokontroler

Berikut adalah barisan bahasa pemrograman menggunakan Android Development Tools untuk fungsi menutup sambungan dengan perangkat mikrokontroler :

```
void closeBT() throws IOException {
    try {
        stopWorker = true;
        mmOutputStream.close();
        mmInputStream.close();
        mmSocket.close();

        Toast.makeText(getApplicationContext(), "HAS Bluetooth Terminated", Toast.LENGTH_SHORT).show();
        status.setText("Press exit button");
        receive.setText("Disconnected");
        toggle1.setEnabled(false);
        toggle2.setEnabled(false);
        toggle3.setEnabled(false);
        toggle4.setEnabled(false);
        toggle5.setEnabled(false);
        toggleall.setEnabled(false);
```

```
                closeButton.setEnabled(false);
                openButton.setEnabled(true);
                exitButton.setEnabled(true);
            } catch (NullPointerException e) {
                e.printStackTrace();
            } catch (Exception e) {
                e.printStackTrace();
            }
    }
```

4.3.4 Tampilan pada Android

Untuk dapat mengendalikan Mikrokontroler melalui Bluetooth maka penulis menggunakan Tampilan pada Android. dengan cara mengirimkan paket-paket paket data perintah. Pada halaman selanjutnya adalah beberapa *User Interface* pada Android.

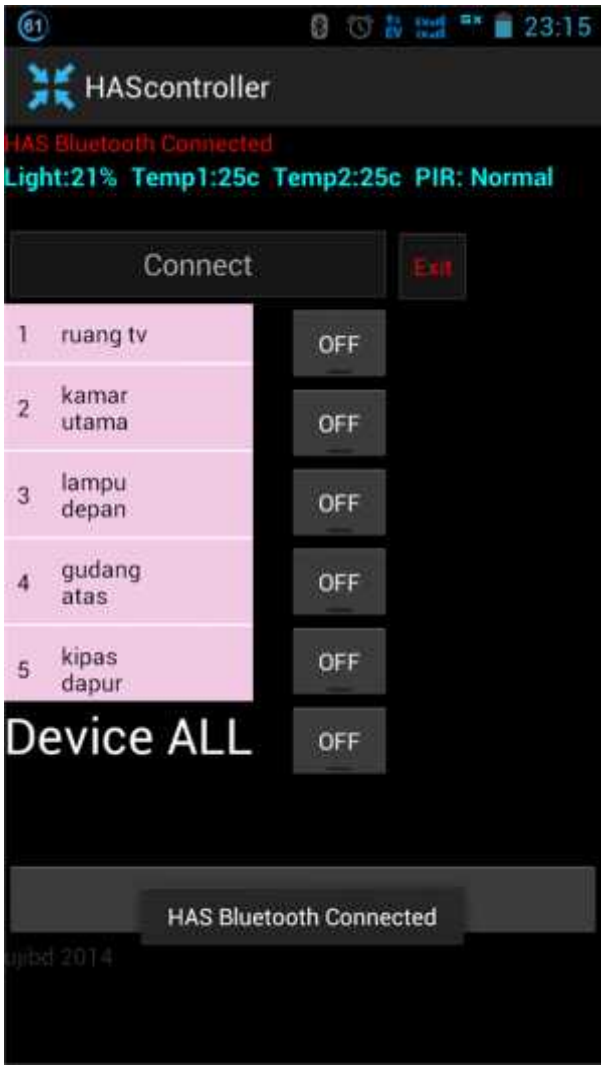
4.3.4.1 Kondisi Awal Pogram



Gambar 4.5 Kondisi Awal Program

Kondisi Awal Program adalah pertama kali membuka program atau program pertama kali dijalankan, pada saat tersebut mikrokontroler belum terhubung.

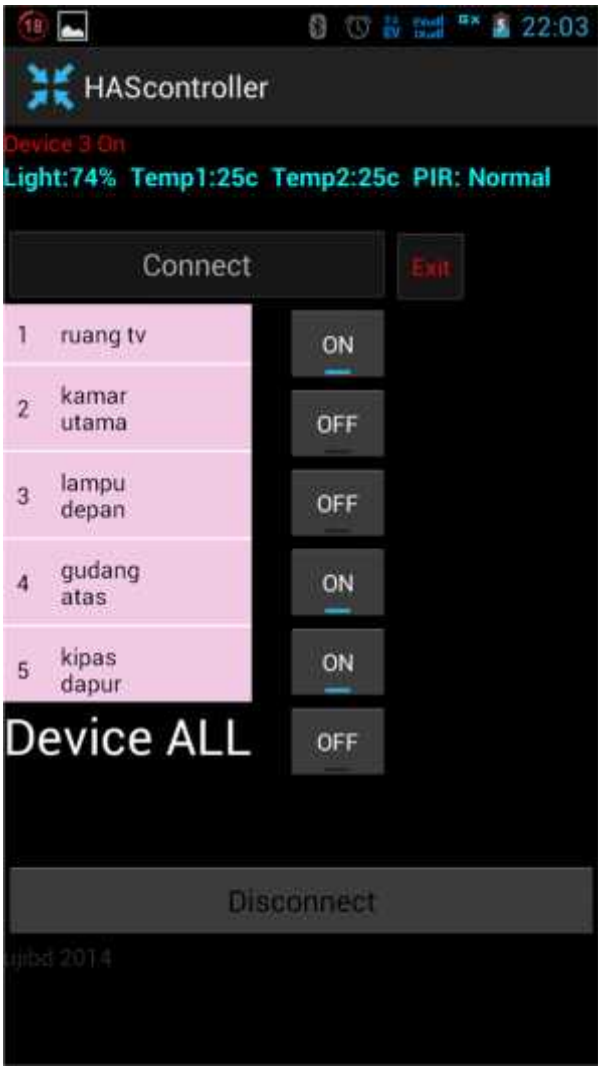
4.3.4.2 Kondisi Program Dihidupkan dan sudah terhubung dengan perangkat



Gambar 4.6 Kondisi Program Dihidupkan dan terhubung dengan perangkat

Kondisi Program Dihidupkan Tanpa Pemakaian Listrik adalah kondisi pada saat program sudah terhubung dengan perangkat Sistem Rumah Pintar Sederhana

4.3.4.3 Kondisi Program Dihidupkan dan Adanya Device yang dihidupkan



Gambar 4.7 Kondisi Program Dihidupkan dan Adanya Device yang dihidupkan

Kondisi Program Dihidupkan Dengan Adanya Device yang Dihidupkan adalah kondisi dimana aplikasi sudah terhubung dimana arus listrik telah didistribusikan dan adanya pemakaian arus listrik.

4.3.4.5 Kondisi Program Dihidupkan dan Hubungan dengan perangkat diputus



Gambar 4.8 Kondisi Program Dihidupkan dan Hubungan dengan perangkat diputus

Kondisi Program Dihidupkan dan Hubungan dengan perangkat diputus adalah dimana program sudah terhubung dengan perangkat mikrokontroler kemudian pengguna menekan tombol disconnect.

4.4 Pengujian dan Analisa Data

Metode pengujian pada penelitian ini dimulai dengan pengujian Mikrokontroler ATmega8. Perlu adanya pengujian mikrokontroler agar kita tahu bahwa mikrokontroler bekerja dengan baik atau tidak kerena hal ini sangat penting mengingat pengintegrasi semua system utama adalah mikrokontroler ATmega8. Selanjutnya yang diuji adalah sensor. Ada empat buah sensor yang akan diuji nantinya adalah sensor gerak (PIR), Sensor cahaya (LDR), Sensor Suhu 1 dan sensor suhu 2 serta fungsi untuk menghidupkan atau mematikan perangkat elektrik pada android.

4.4.1 Pengujian Minimum sistem ATmega8

Tujuan dari pengujian mikrokontroler adalah untuk mengetahui bahwa setiap port yang digunakan pada mikrokontroler ATmega8 bekerja dengan baik dengan nilai tegangan 4.5-5 Volt. Pengujian khususnya dilakukan pada Port yang digunakan. Pengujian dilakukan dengan cara menghubungkan probe positif AVO meter ke Port yang diuji dan probe Negatif pada Ground.

Hasil yang didapat adalah 4.9 Volt yang artinya adalah logika “1” yang mengindikasikan bahwa minimum sistem ATmega8 bekerja dengan baik.

4.4.2 Pengujian Sensor gerak (PIR)

Pengujian sensor PIR dilakukan dengan melihat sensitifitas sensor terhadap jarak tempuh. Hasilnya adalah pada jarak kurang dari sama dengan 5 meter sensor ini dapat bekerja dan apabila lebih dari 5 meter maka tidak akan terdeteksi karena PIR bukan hanya mengamati pergerakan manusia saja, akan

tetapi juga pancaran sinar inframerah pasif dari tubuh manusia. Datanya adalah sebagai berikut:

Table 4.1 Data Pengujian Sensitifitas PIR terhadap Jarak

Jarak	Intensitas kepekaan PIR terhadap Objek	
	Terdeteksi	Tidak Terdeteksi
1 Meter	✓	
2 Meter	✓	
3 Meter	✓	
4 Meter	✓	
5 Meter	✓	
6 Meter		✓
7 Meter		✓

4.4.3 Pengujian pada Sensor Cahaya (Light Dependant Resistor)

Pengujian dilakukan dengan mengukur tegangan yang ada pada kaki LDR dengan suatu kondisi intensitas cahaya tertentu dengan cara sensor ditutupi dan disorot oleh senter LED. Berikut ini adalah hasil pengukuran intensitas cahaya terhadap tegangan:

Table 4.2 Data Pengujian Light Dependant Resistor (LDR)

Intensitas Cahaya	Tegangan (Volt)	Nilai pada LCD & Android (%)
Ditutupi (Gelap)	0.6	0 -6
Tersorot (Cerah)	1.3	90-100

Semakin besar cahaya yang diterima maka akan semakin kecil nilai resistansi yang ada pada LDR. Hal ini disebabkan karena LDR terbuat dari bahan semikonduktor yang terbuat dari cadmium sulfide (CdS) dan cadmium seleneida (CdSe) yang mana bahan-bahan ini sangat sensitive terhadap cahaya.

4.4.4 Pengujian pada Sensor Suhu 1 dan Sensor Suhu 2

Pengujian Sensor suhu 1 dan 2 dilakukan dengan membandingkan dengan thermometer digital (kalibrator), pengujian dilakukan pada waktu & tempat yang berbeda. Dalam pengujian ini terdapat beberapa persen kesalahan yang masih dapat ditoleransikan yang dikarenakan proses pelepasan panas dari sensor suhu membutuhkan waktu sehingga terjadinya keterlambatan perubajan suhu. Dari pengujian didapat hasil data sebagai berikut:

Table 4.3 Data Pengujian Sensor Suhu 1

Waktu menit Ke-	Suhu (Celcius)		Error (%)
	Kalibrator	Display LCD & Android	
1	25	25	0
2	25	25	0
3	24	25	4.16
4	25	25	0
5	24	25	4.16
6	24	24	0
7	24	24	0
8	24	24	0
9	25	24	4
10	25	25	0
Rata-Rata	24.5	24.6	1.23

Table 4.4 Data Pengujian Sensor Suhu 2

Waktu menit Ke-	Suhu (Celcius)		Error (%)
	Kalibrator	Display LCD & Android	
1	23	23	0
2	23	23	0
3	23	23	0
4	23	23	0
5	25	24	4.16
6	25	25	0
7	25	25	0
8	23	25	8.69
9	23	24	4.34
10	23	23	0
Rata-Rata	24.5	24.6	1.71

Percentage Error dari Sensor Suhu 1 dan 2 dihitung menggunakan rumus sebagai berikut:

$$\%Error = \left(\frac{Nilai\ Kalibrator - Nilai\ Digital}{Nilai\ Kalibrator}\right) \times 100\,\%$$

4.4.5 Pengujian fungsi mengendalikan perangkat elektrikal pada android

Pengujian fungsi pada android dilakukan dengan menjalan aplikasi *"HAS Controller"* pada *smartphone* android dan sudah terhubung pada perangkat Sistem Rumah Pintar Sederhana dengan penghubung atau media komunikasi menggunakan *Bluetooth.* Datanya adalah sebagai berikut (pada halaman selanjutnya):

Table 4.5 Pengujian fungsi pada android

Fungsi pada Perangkat		Berfungsi?
Android	Sistem Rumah Pintar Sederhana	
Device 1 ON	Load 1 ON	Ya
Device 2 ON	Load 2 ON	Ya
Device 3 ON	Load 3 ON	Ya
Device 4 ON	Load 4 ON	Ya
Device 5 ON	Load 5 ON	Ya
All Device ON	ALL Load ON	Ya
Device 1 OFF	Load 1 OFF	Ya
Device 2 OFF	Load 2 OFF	Ya
Device 3 OFF	Load 3 OFF	Ya
Device 4 OFF	Load 4 OFF	Ya
Device 5 OFF	Load 5 OFF	Ya
All Device OFF	ALL Load OFF	Ya