

## **BAB III**

### **ANALISA SISTEM DAN DESIGN**

Pada Bagian ini akan dibahas lebih lanjut mengenai pembuatan dan penerapan beberapa dari landasan teori yang ada untuk menjawab rumusan masalah yang dipaparkan pada Bab I.

#### **3.1 Software Development Life Cycle**

Dalam merancang Sistem Otomatisasi Rumah ini, penulis menggunakan metode Software Development Life Cycle untuk menyelesaikan Sistem tersebut. Berdasar referensi, *Classical and Object-Oriented Software Engineering with UML and Java, Chapter 3*, Software Development Life Cycle menjadi beberapa fase yaitu :

##### *1.1.1 Requirement*

Dilakukan oleh si penulis untuk mengetahui apa keinginan user akan Sistem Otomatisasi Rumah yang dirancang dengan metode menyebar Kuisisioner dengan jumlah 100 sampel ke area Surabaya dan sekitarnya.

##### *1.1.2 Analyze*

Dilakukan oleh si penulis ketika hasil kuisisioner yang telah diedarkan tersebut selesai dikumpulkan datanya. Di sini penulis menganalisa data-data tersebut untuk memperoleh kebutuhan user akan Sistem Otomatisasi Rumah seperti apakah yang baik dan diharapkan oleh end user nantinya.

### *1.1.3 Design and Implementation*

Pada tahap ini si penulis membuat system architecture dari Sistem Otomatisasi Rumah yang hendak dirancang terlebih dahulu mulai dari Use Case Diagram, Interface Flow Diagram, dan ER Diagram dari Sistem Otomatisasi Rumah tersebut baru kemudian penulis mulai merancang kode-kode program dari Sistem Otomatisasi Rumah tersebut.

### *1.1.4 User Testing*

Setelah Sistem Otomatisasi Rumah telah selesai dirancang oleh penulis, harapannya penulis dapat mengujicobakan Sistem tersebut langsung kepada end user. Namun karena Sistem Otomatisasi Rumah yang dirancang oleh penulis membutuhkan waktu yang cukup panjang untuk diterapkan langsung di rumah pengguna, maka User Testing hanya bisa dilakukan sebatas maket dengan ukuran 40cm X 80cm dengan menggunakan LED.

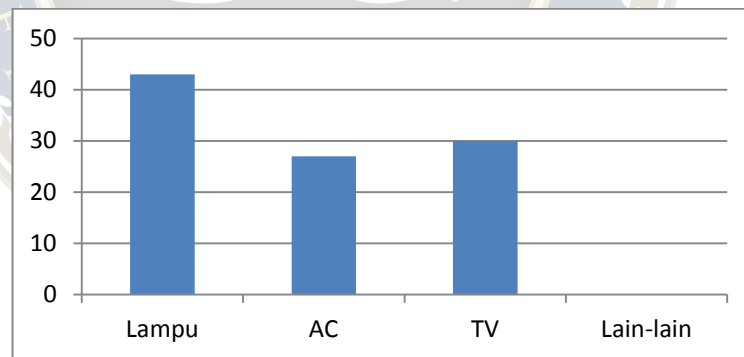
### *1.1.5 Integration and Maintenance*

Setelah Sistem Otomatisasi Rumah berhasil diterapkan, perlu dilakukan proses integrasi dan maintenance terhadap kinerja Sistem untuk memperbaiki adanya kesalahan Sistem atau kesalahan prosedur penggunaan Sistem.

### 3.2 Preliminary Research

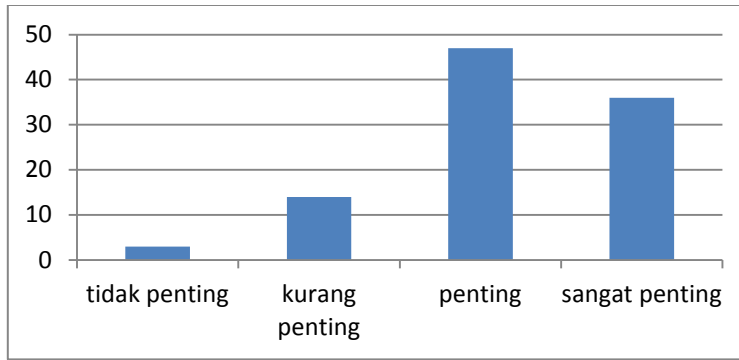
Preliminary Research yang dilakukan adalah dengan melakukan survey yang disebarkan ke seratus sampel yang berada di area Surabaya dan sekitarnya dengan tujuan untuk menguji kelayakan dari manfaat Sistem Otomatisasi Rumah yang akan dirancang. Lalu bagaimanakah Sistem Otomatisasi Rumah yang diharapkan oleh orang-orang tersebut (fitur-fitur apakah yang diharapkan). Hasil dari Preliminary Research oleh 100 sampel tersebut adalah demikian :

#### 1. Alat Listrik yang Sering untuk manfaat penghematan



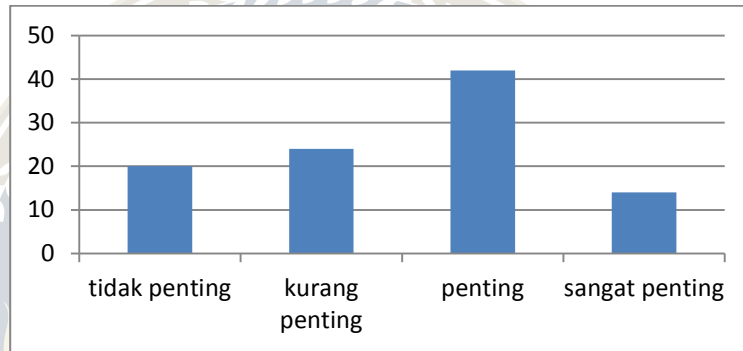
*Gambar 3.2.1 Survei Alat Listrik Untuk Penghematan*

#### 2. Pentingnya Penghematan Listrik



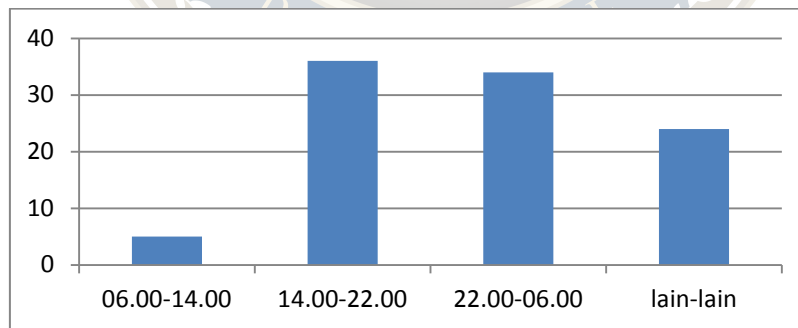
Gambar 3.2.2 Survei Pentingnya Penghematan Listrik

### 3. Pentingnya Waktu Nyala dan Mati Alat Listrik



Gambar 3.2.3 Survei Pentingnya Waktu Nyala dan Mati Alat Listrik

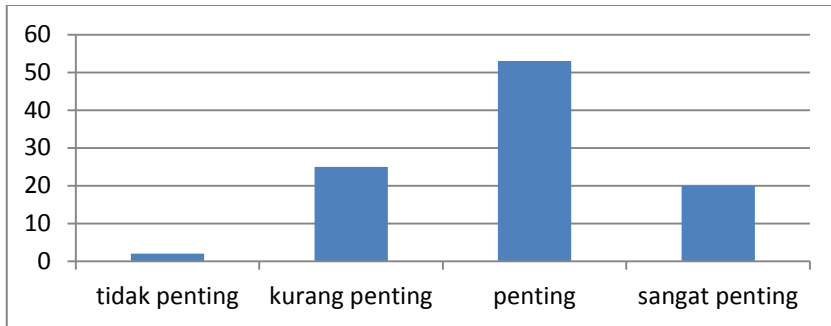
### 4. Range Waktu Alat Listrik Dalam Kondisi Nyala



Gambar 3.2.4 Survei

Range Waktu Nyala

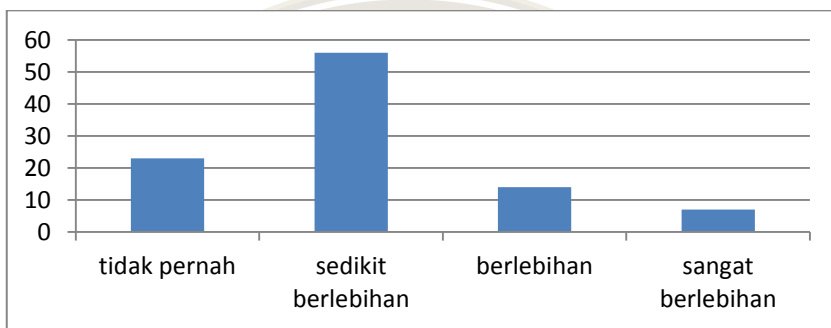
### 5. Pentingnya Informasi Biaya Listrik



Gambar 3.2.5

*Survei Pentingnya Informasi Biaya Listrik*

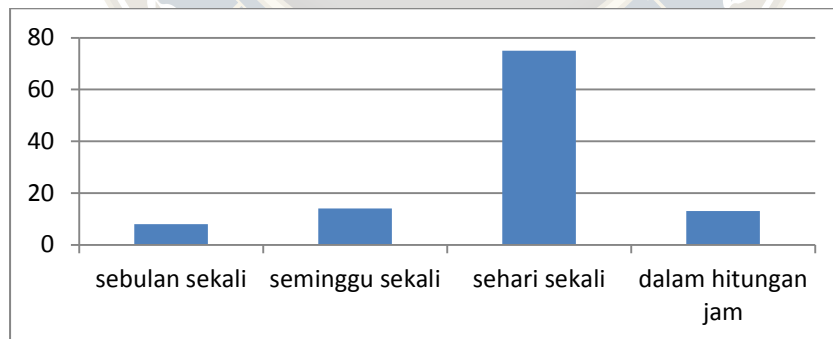
## 6. Seringnya Pemborosan Listrik Di Rumah



Gambar 3.2.6

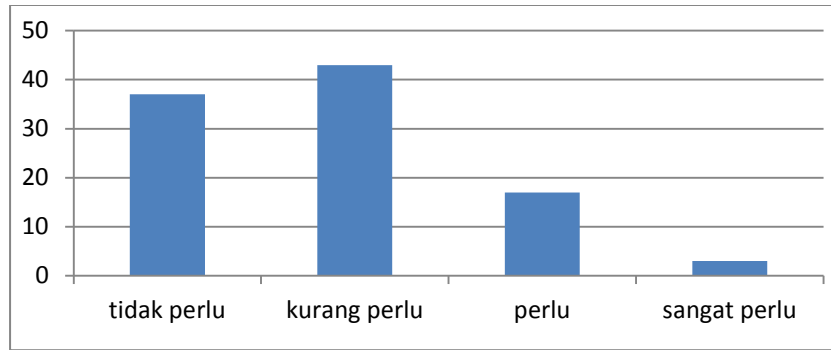
*Survei Seringnya Pemborosan Alat Listrik*

## 7. Rutinitas Pemakaian Alat Listrik



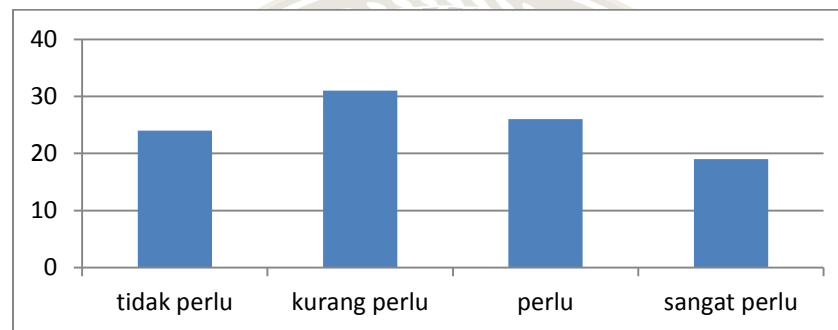
Gambar 3.2.7 Survei Rutinitas Pemakaian Alat

## 8. Perlunya Reminder Untuk Pemakaian Listrik (Ketika user telah menyalakan alat listrik tanpa penjadwalan)



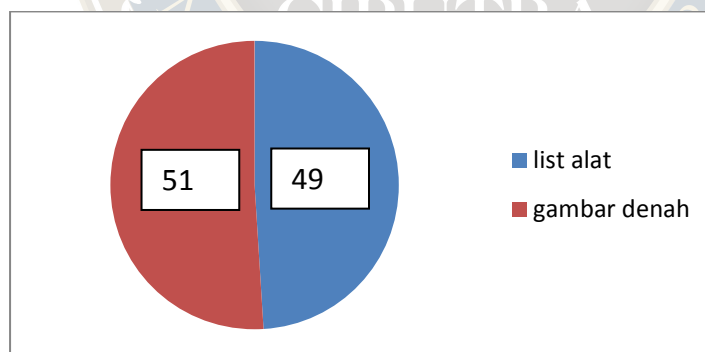
Gambar 3.2.8 Survei Perlunya Reminder

### 9. Perlunya Opsi Pilihan Waktu Untuk Mematikan Alat Listrik



Gambar 3.2.9 Survei Opsi Pilihan Waktu

### 10. UI Yang Diharapkan



Gambar 3.2.10 Survei UI

Dari hasil preliminary research oleh 100 sampel tersebut dapat disimpulkan hal-hal berikut ini :

1. **Lampu** adalah alat listrik yang dominan digunakan untuk keperluan penghematan listrik

2. Penghematan listrik memiliki prioritas yang **penting**
3. Pentingnya waktu nyala alat listrik adalah **penting**
4. Range waktu nyala alat listrik dominan pada **14.00-22.00 WIB**
5. Informasi biaya listrik adalah **penting**
6. Seringnya pemborosan listrik dominan pada **sedikit berlebihan**
7. Rutinitas pemakaian alat listrik dominan pada **sehari sekali**
8. Perlunya reminder(pengingat) dominan pada **kurang perlu**
9. Perlunya opsi pilihan waktu untuk mematikan alat yaitu **kurang perlu**
10. UI dari aplikasi yang disukai dominan pada **denah dengan angka dominan 2%**

Dengan demikian manfaat yang diharapkan dari Sistem Otomatisasi Rumah yang dirancang **sangatlah diharapkan** berdasarkan hasil preliminary research tersebut.

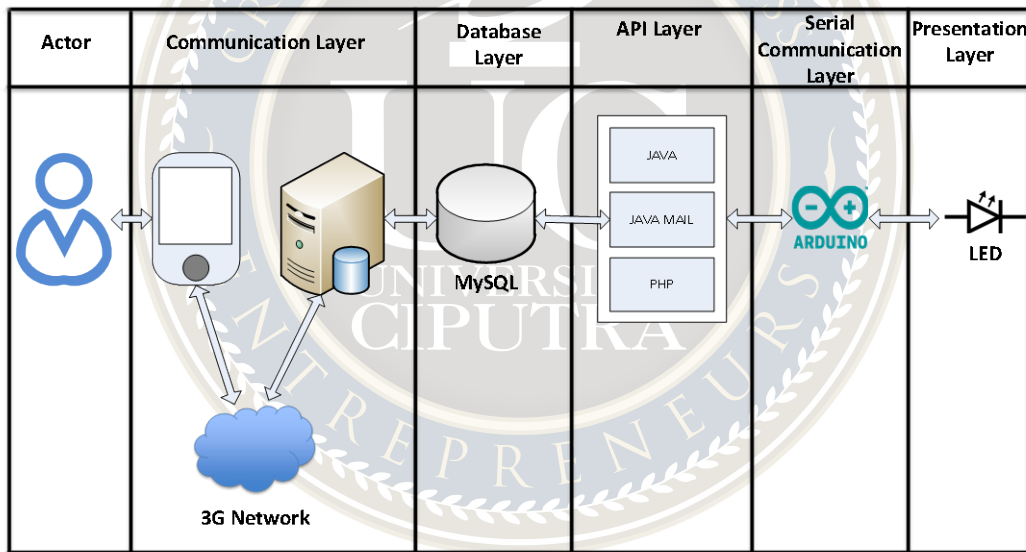
Untuk fitur tambahan yang diinginkan berdasarkan preliminary research tersebut adalah **informasi biaya listrik**.

Untuk fitur yang kurang diinginkan berdasarkan data di atas adalah **sistem pengingat untuk mematikan alat listrik yang dinyalakan tanpa melalui jadwal**.

Ada satu hal khusus yang tidak dilakukan oleh penulis berdasar data dari preliminary research tersebut yaitu, **membuat aplikasi android dengan UI berupa informasi table dari device**. Hal ini dikarenakan tujuan utama penulis untuk memproduksi aplikasi dan sistem ini secara masal. Tidak menutup kemungkinan bagi penulis untuk merancang UI dari aplikasi berupa denah rumah user Sistem Otomatisasi Rumah, akan tetapi dibutuhkan usaha lebih bagi penulis untuk melakukan perancangan UI sesuai denah rumah user, dimana setiap rumah user itu akan memiliki denah rumah yang unik pastinya.

### 3.3 System Architecture

Untuk mempermudah kinerja penulis dalam membuat Sistem Otomatisasi Rumah ini, penulis mempersiapkan rancangan dari sistem arsitektur Sistem Otomatisasi Rumah dirancang sebagai berikut :



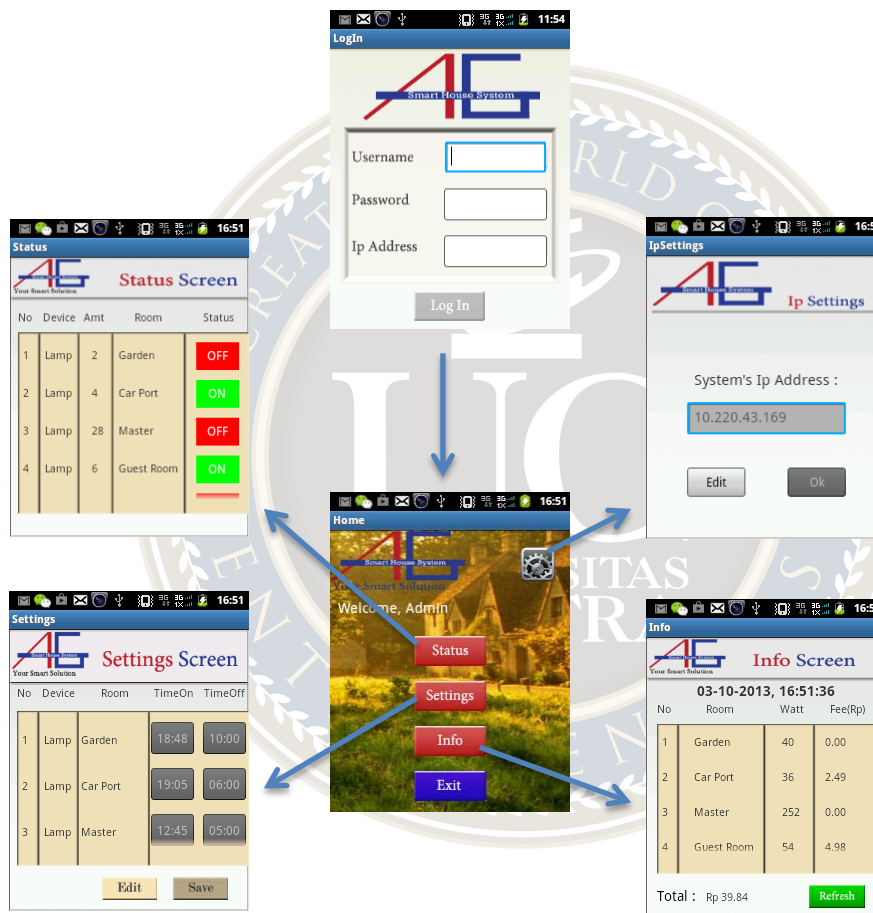
Gambar 3.3 Diagram Arsitektur

Dengan diagram arsitektur yang telah dirancang seperti di atas, maka untuk merancang Sistem Otomatisasi Rumah dapat dilakukan sesuai dengan apa yang dinyatakan oleh notasi-notasi pada diagram arsitektur di atas.

### 3.4 Interface Flow Diagram

Sebelum memulai untuk proses pembuatan aplikasi dari Sistem Otomatisasi Rumah, penulis terlebih dahulu membuat Interface Flow Diagram dari Aplikasi yang hendak dirancangnya sebagai berikut :

#### 3.4.1 Android Application



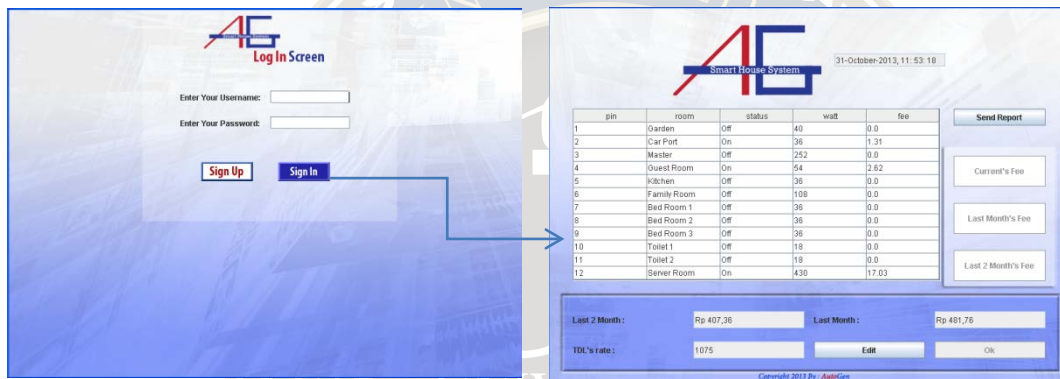
Gambar 3.4.1 Android Application Interfaces Flow

#### 3.4.2 Java Desktop Application





Gambar 3.4.2.1 Java Application Interface Flow 1

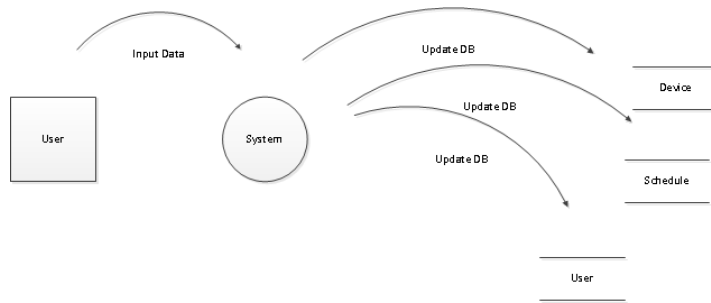


Gambar 3.4.2.2 Java Application Interface Flow 2

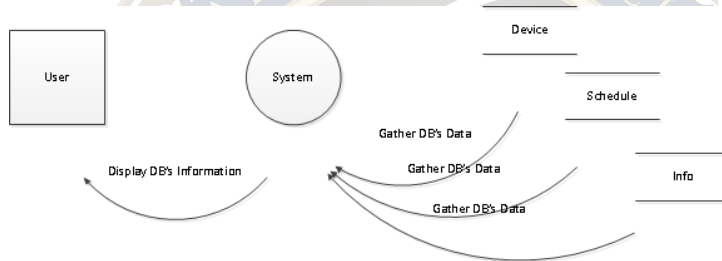
### 3.5 Data Flow Diagram

Berikut Data Flow Diagram dari Sistem Otomatisasi Rumah yang dirancang :

#### 3.5.1 Level 0 DFD

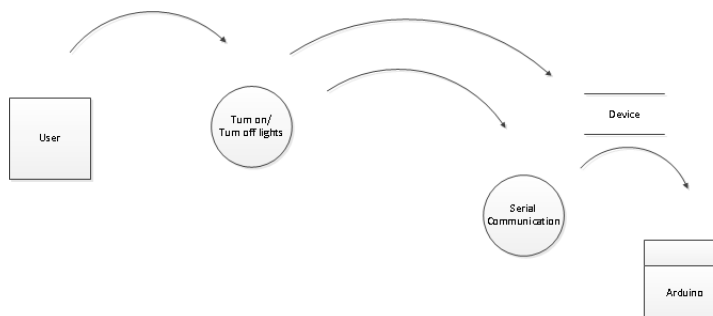


*Gambar 3.5.1.1*  
 User's Input/ Update Schedule,  
 device's status, and make a new user

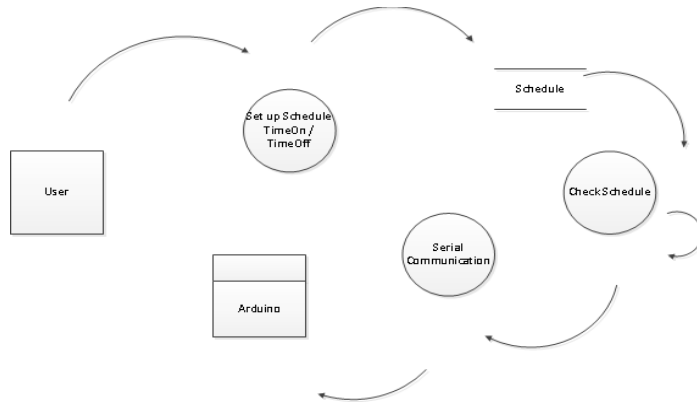


*Gambar 3.5.1.2*  
 User's is viewing Device,  
 Schedule, and Info DB

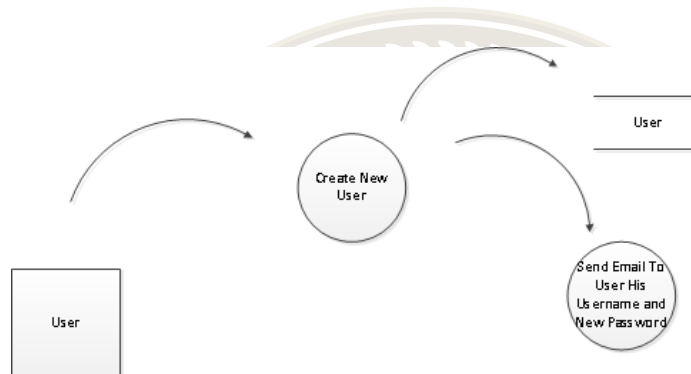
### 3.5.2 Level 1 DFD



*Gambar 3.5.2.1*  
 Turn On/Off light  
 without scheduling



Gambar 3.5.2.2  
Turn On/Off light  
with scheduling

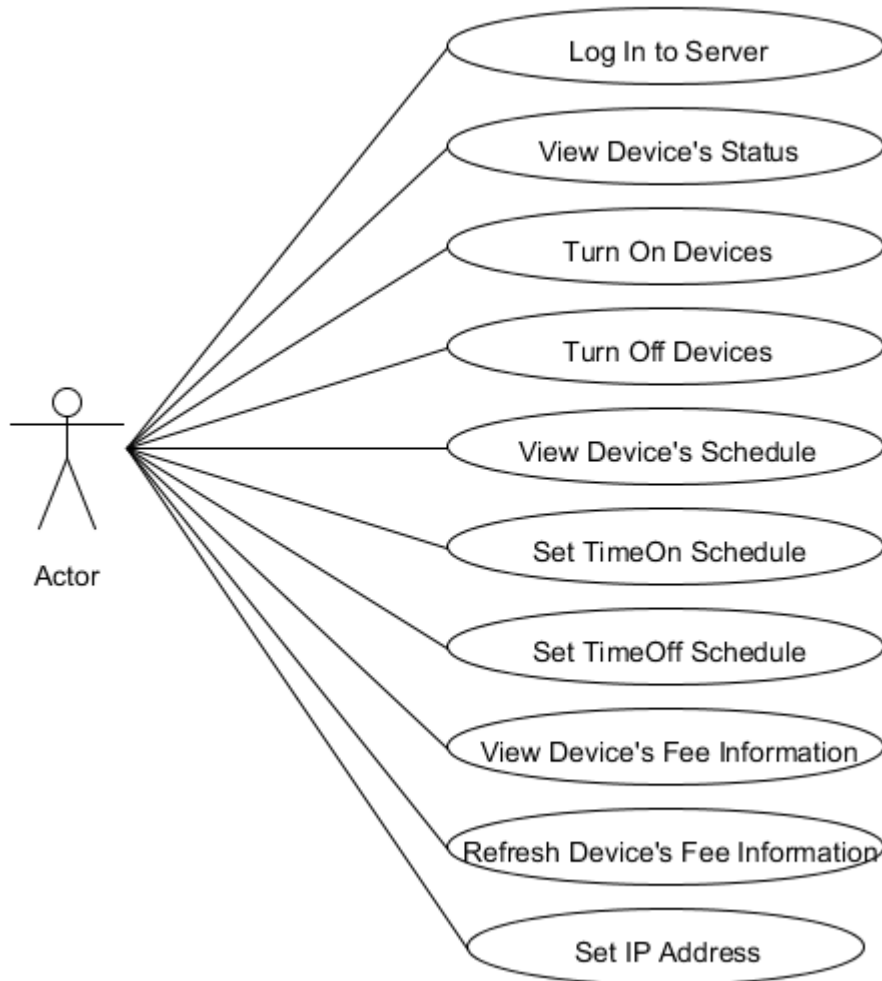


Gambar 3.5.2.3  
Make a New user

### 3.6 Use Case Diagram

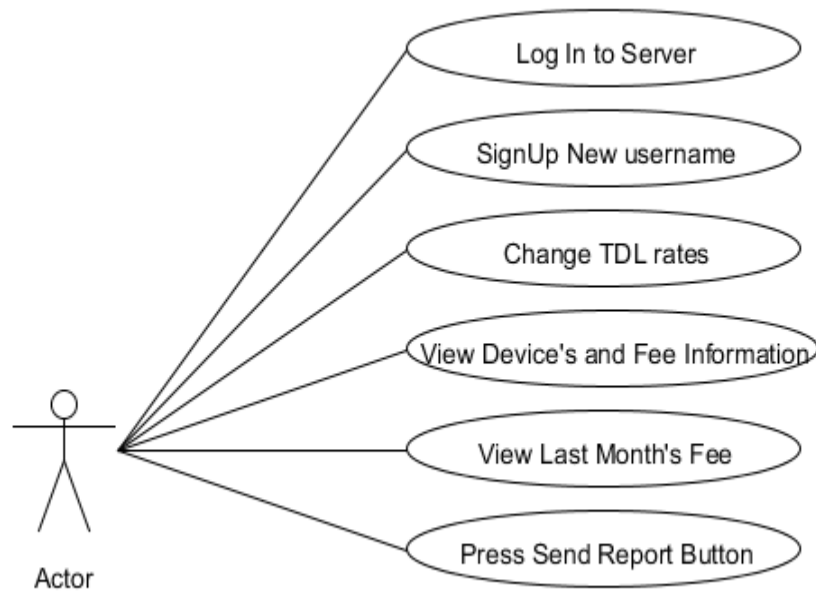
Use Case dari Sistem Otomatisasi Rumah ini adalah sebagai berikut :

### 3.6.1 Android Application



*Gambar 3.6.1 Use Case Android Application*

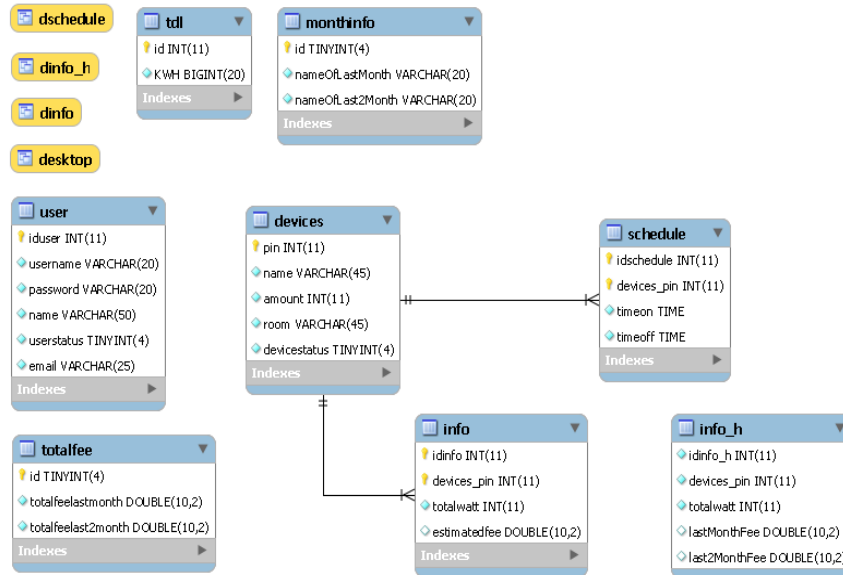
### 3.6.2 Java Desktop Application



Gambar 3.6.2 Use Case Java Application

### 3.7 ER Diagram

ER Diagram dari Sistem Otomatisasi Rumah ini dibuat oleh penulis berdasar referensi *MySQL Tutorial Basic 2013* yang diakses melalui “<http://www.mysqltutorial.org/basic-mysql-tutorial.aspx>” adalah sebagai berikut :



Gambar 3.7 ER Diagram

### 3.7.1 Tabel :

Tools yang digunakan untuk pembuatan tabel-tabel ini adalah MySQL WorkBench dimana untuk proses pembuatannya si penulis melakukan drag and drop tabel yang disediakan oleh aplikasi dan mengisi setiap entitasnya dengan nilai-nilai yang sesuai. Ketika tabel-tabel yang diinginkan telah selesai dibuat, si penulis melakukan *Synchronize Model* agar database yang telah dibuat di-update oleh aplikasi dan disimpan ke dalam database mesin(localhost). Berikut tabel-tabel database yang diperlukan untuk pembuatan Sistem Otomatisasi Rumah :

#### 3.7.1.1 TDL

Tabel TDL ini dibuat dengan tujuan untuk menyimpan nilai tarif TDL dari tarif listrik PLN. Tabel ini berisikan entitas sebagai berikut :

1. Id yang tipenya **INT**
2. KWH(nilai TDLnya) yang tipenya **DOUBLE**

### 3.7.1.2 monthinfo

Tabel monthinfo ini dibuat dengan tujuan untuk menyimpan variabel nama bulan lalu dan nama 2 bulan lalu yang nantinya akan digunakan ketika membuat laporan biaya listrik bulanan. Tabel ini berisikan entity sebagai berikut :

1. Id yang tipenya **INT**
2. nameOfLastMonth yang tipenya **VARCHAR**
3. nameOfLast2Month yang tipenya **VARCHAR**

### 3.7.1.3 user

Tabel user dibuat untuk menyimpan data user-user dari Sistem Otomatisasi Rumah itu sendiri. Tabel user ini berisi entity sebagai berikut :

1. id yang tipenya **INT**
2. username yang tipenya **VARCHAR**
3. password yang tipenya **VARCHAR**
4. name yang tipenya **VARCHAR**
5. userstatus yang tipenya **TINYINT**
6. email yang tipenya **VARCHAR**

### 1.7.1.4 totalfee

Tabel totalfee ini digunakan untuk menyimpan biaya listrik total bulan lalu dan 2 bulan yang lalu. Tabel totalfee ini berisi entity sebagai berikut :

- 2 id yang tipenya **INT**
- 3 totalfeelastrmonth yang tipenya **DOUBLE**
- 4 totalfeelastr2month yang tipenya **DOUBLE**

#### 3.7.1.5 devices

Tabel devices digunakan untuk menyimpan data-data dari device yang terdaftar di dalam Sistem Otomatisasi Rumah yang akan dirancang.

Tabel devices memiliki entity sebagai berikut :

1. pin yang tipenya **INT**
2. name yang tipenya **VARCHAR**
3. amount yang tipenya **INT**
4. room yang tipenya **VARCHAR**
5. status yang tipenya **TINYINT**

#### 3.7.1.6 schedule

Tabel schedule digunakan untuk menyimpan jadwal nyala dan mati devices yang telah didaftarkan pada database Sistem Otomatisasi Rumah yang akan dirancang. Tabel schedule memiliki entity sebagai berikut :

1. id yang tipenya **INT**
2. devices\_pin yang merupakan *Foreign Key* dari entity pin tabel *devices*
3. timeon yang tipenya **TIME**
4. timeoff yang tipenya **TIME**

### 3.7.1.7 info

Tabel info digunakan untuk menyimpan informasi devices lebih lanjut tentang watt device, dan biaya listrik device tersebut. Tabel info memiliki entity sebagai berikut :

1. id yang tipenya **INT**
2. devices\_pin yang merupakan *Foreign Key* dari entity pin tabel

*devices*

3. totalwatt yang tipenya **INT**
4. estimatedfee yang tipenya **DOUBLE** dengan *decimal places*

*yaitu 2*

### 3.7.1.8 info\_h

Tabel info\_h digunakan untuk menyimpan history dari biaya listrik per device pada bulan lalu dan 2 bulan lalu. Isi dari entity akan diupdate ketika akan memasuki awal bulan tanggal 1. Entity yang dimiliki tabel ini hampir sama dengan tabel info, namun terdapat perbedaan yaitu entity lastMonthFee yang tipenya **DOUBLE** dan last2MonthFee yang tipenya **DOUBLE**.

### 3.7.2 View :

Untuk mempermudah penulis untuk mengambil data dari beberapa tabel di atas yang telah dibuat tersebut, maka penulis membuat sebuah View yang

dapat diakses untuk menampilkan data-data dari beberapa tabel yang diperlukan. Berikut View-View yang digunakan penulis untuk mengakses data pada Sistem Otomatisasi Rumah yang akan dirancang

### *3.7.2.1 desktop*

View ini nantinya akan digunakan untuk menampilkan data-data pada aplikasi desktop dari Sistem Otomatisasi Rumah yang akan dirancang. Query yang akan dibutuhkan untuk membuatnya yaitu :

```
CREATE
VIEW desktop AS
select
    `i`.`devices_pin` AS `pin`,
    `d`.`room` AS `room`,
    `d`.`devicestatus` AS `status`,
    `i`.`totalwatt` AS `watt`,
    `i`.`estimatedfee` AS `fee`
from
    (`info` `i`
    join `devices` `d` ON ((`d`.`pin` = `i`.`devices_pin`)))
```

### *3.7.2.2 dinfo*

View ini digunakan untuk menampilkan info dari setiap alat listrik dengan menggabungkan tabel devices dan info. Query yang diperlukan untuk membuatnya yaitu :

```
CREATE
```

```
VIEW `dinfo` AS
```

```
select
```

```
  `i`.`devices_pin` AS `pin`,
```

```
  `d`.`room` AS `room`,
```

```
  `i`.`totalwatt` AS `watt`,
```

```
  `i`.`estimatedfee` AS `fee`
```

```
from
```

```
  (`info` `i`
```

```
  join `devices` `d` ON ((`d`.`pin` = `i`.`devices_pin`)))
```

### 3.7.2.3 *dinfoh*

View ini digunakan untuk menampilkan info\_h yang digabung dengan tabel devices yang nantinya akan digunakan untuk keperluan pengiriman laporan biaya listrik melalui email. Query yang diperlukan untuk membuatnya yaitu :

```
CREATE
```

```
VIEW `dinfo_h` AS
```

```
select
```

```
  `i`.`devices_pin` AS `pin`,
```

```
  `d`.`room` AS `room`,
```

```
  `i`.`totalwatt` AS `watt`,
```

```
  `i`.`lastMonthFee` AS `lastMonthFee`,
```

```
  `i`.`last2MonthFee` AS `last2MonthFee`
```

```
from
```

```
(`info_h` `i`  
join `devices` `d` ON ((`d`.`pin` = `i`.`devices_pin`)))
```

#### 3.7.2.4 dschedule

View ini digunakan untuk menampilkan jadwal-jadwal dari tiap device yang terdaftar dengan menggabungkan tabel schedule dan devices yang nantinya digunakan dalam aplikasi Android. Query yang diperlukan untuk membuat view ini adalah :

```
CREATE  
VIEW `dschedule` AS  
select  
  `s`.`devices_pin` AS `pin`,  
  `d`.`name` AS `name`,  
  `d`.`room` AS `room`,  
  `s`.`timeon` AS `timeon`,  
  `s`.`timeoff` AS `timeoff`  
from  
  (`schedule` `s`  
  join `devices` `d` ON ((`d`.`pin` = `s`.`devices_pin`)))
```

### 3.8 Skema Elektrik

Penulis merancang skema elektrik untuk menghubungkan rangkaian proto-type yang akan diimplementasikan ke maket dengan menggunakan referensi buku *Teach Yourself Electricity and Electronics : fourth edition. McGraw-Hill.*

Di buku itu dijelaskan beberapa kualifikasi dan teori-teori dasar mengenai perancangan sirkuit elektronik yang baik dan bagaimana resistansi dari setiap kabel-kabel yang digunakan dalam rangkaian berpengaruh terhadap alat listrik tersebut.

Namun tidak semua teori di buku tersebut digunakan dan diimplementasikan dalam proses pembuatan Sistem Otomatisasi Rumah yang kan dirancang penulis.

Beberapa hal yang diperhatikan oleh penulis ketika membuat rangkaian elektronik yaitu :

1. **Rangkaian parallel** : digunakan untuk memecah voltase ruangan yang berbeda
2. **Rangkaian seri** : digunakan untuk menghubungkan voltase alat listrik pada ruangan yang sama
3. Batas-batas **Voltase** dan **daya** beserta **resistansi** dari beberapa alat listrik untuk demo seperti LED, relay, dsb.

Hal-hal di atas dijelaskan lebih lanjut dengan jelas di dalam buku acuan yang digunakan oleh penulis.

### **3.9 Komunikasi Mikronkontroler ke Device dan PC**

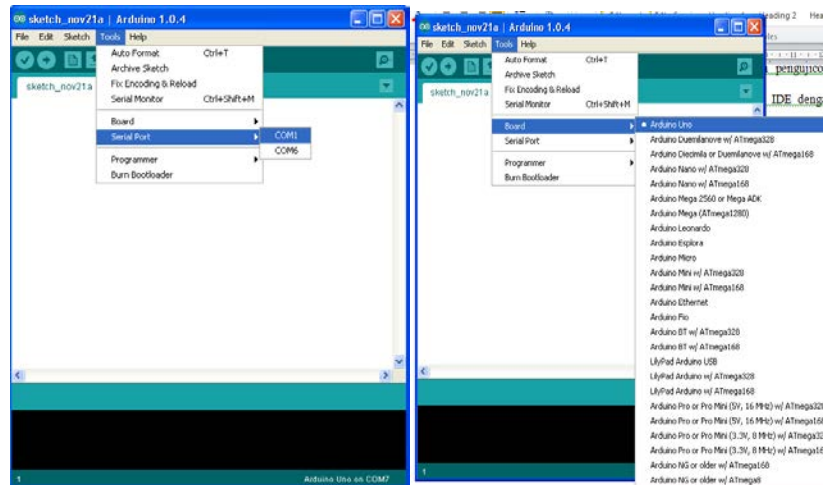
Pada bagian ini akan dibahas metode-metode atau langkah-langkah untuk mengkomunikasikan mikrokontroler dengan jenis Arduino Uno ke Device(alat listrik) dan PC.

#### **3.9.1 Uji Coba Serial Connection**

Untuk melakukan hal ini, langkah pertama yang dilakukan oleh penulis adalah menghubungkan Arduino yang berperan sebagai mikrokontroller ke PC melalui USB Serial. Pada mulanya, Arduino Uno pasti tidak akan dikenali oleh

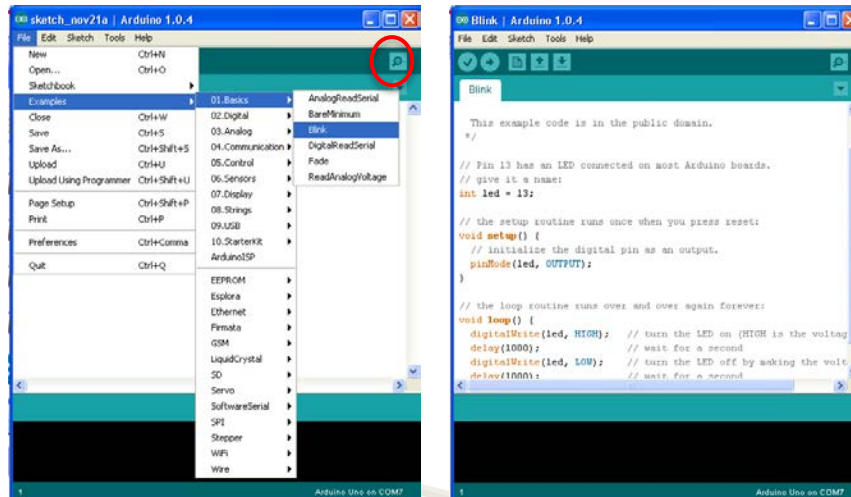


Arduino IDE dengan memilih parameter yang diperlukan pada Arduino IDE tersebut yaitu jenis COM Port dimana Arduino Uno tersebut tercolok dan jenis boardnya sendiri yaitu UNO.



Gambar 3.9.1.3 Parameter Setup Arduino Uno

Setelah berhasil, dapatlah diujicobakan sample code yang telah disediakan oleh IDE Arduino misal : Blink.ino . Buka sample code tersebut, dan unggah code tersebut pada Arduino Uno yang masih tercolok pada USB Serial. Jikalau sukses, dapat dilihat LED disebelah pin nomor 13 board Arduino Uno akan berkedip.



Gambar 3.9.1.4 Uploading Blink ke Arduino Uno

### 3.9.2 Direct Connection dari Arduino Uno ke Device

Setelah berhasil mengujicoba Serial Connection seperti pada bagian sebelumnya, dapat diketahui bahwa Arduino Uno yang telah tersambung ke PC sudah siap digunakan.

Selanjutnya untuk mengujicobakan koneksi alat ke Arduino, hal pertama yang perlu dilakukan adalah melakukan pengamatan pada digital input yang terletak di sebelah kanan board Arduino Uno yang dapat dilihat urutannya mulai dari nomor 0 hingga 13 yang muatan dari pin-pin tersebut akan selalu bernilai positive dan di atas nomor 13 tersebut terdapat pin bernama ground yang muatan dari pin ground tersebut akan selalu bernilai negative.

Pada bagian sebelumnya, diketahui bahwa sample code Blinking LED berfungsi untuk memberikan effect kedipan di pin nomor 13, namun pada pin nomor 13 masih belum ada alat yang tersambung, hanya LED indicator saja yang menunjukkan bahwa pin nomor 13 sudah hidup.

Selanjutnya, penulis mengambil 2 buah kabel jumper dengan warna berbeda, katakanlah warna hitam dan merah serta penulis mengambil sebuah LED berwarna putih. Penulis selanjutnya menghubungkan kabel jumper berwarna hitam yang telah dikupas ujung kulitnya hingga terlihat tembaganya ke pin ground pada board Arduino Uno dan kabel jumper berwarna merah yang telah dikupas ujung kulitnya juga ke pin nomor 13 pada board Arduino Uno. Pada bagian ujung lain dari masing-masing kabel tersebut, penulis menghubungkannya dengan kaki LED dimana kaki LED yang lebih panjang(bernilai positive) dihubungkan dengan kabel berwarna merah, dan kaki LED yang lebih pendek(bernilai negative) dihubungkan dengan kabel berwarna hitam.

Masih pada sample code Blinking LED yang masih terunggah pada chip AtMega board Arduino Uno, jikalau hal yang telah dilakukan penulis di atas telah berhasil maka, LED tersebut akan memberikan sinar berwarna putih dengan effect berkedip dan Direct Connection dari Alat ke Arduino Uno dapat dinyatakan telah berhasil.

### **3.9.3 Serial Communication dari PC ke Alat**

Setelah berhasil melakukan 2 hal di atas, dilakukan hal lebih lanjut untuk menghidupkan atau mematikan sebuah LED di nomor 13 tersebut dengan menginputkan huruf melalui keyboard pada PC.

Masih pada rangkaian yang sama seperti uji coba di bagian sebelumnya, penulis mencoba untuk membuat code yang nantinya diunggah ke chip AtMega board Arduino Uno dengan effect yang berbeda. Berikut contoh code yang digunakan oleh penulis :

```

void setup() {
  Serial.begin(9600); // initialize serial communication:
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, 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
        Serial.println("HIGH");
        break;
      case 'v':
        digitalWrite(13, LOW); // set the LED off
        Serial.println("LOW");
        break;
    }
  }
}

```

Tujuan penulisan code di atas yaitu ketika penulis membuka aplikasi Hyper Terminal yang berfungsi untuk melakukan serial communication dengan Arduino Uno, penulis dapat mengirim pesan kepada board Arduino Uno berupa char huruf. Jikalau pesan yang dikirimkan oleh penulis ke board Arduino Uno adalah huruf 'u' maka pin nomor 13 tersebut akan hidup dan secara otomatis alat

yang terhubung pada pin nomor 13 juga akan hidup pula dan sebaliknya. Karena pada bagian sebelumnya rangkaian LED yang digunakan masih terhubung pada pin nomor 13, maka jikalau hal ini berhasil maka LED tersebut akan dapat dihidupkan atau dimatikan dengan mengirim pesan berupa huruf 'u' atau 'v' tersebut.

Dikarenakan untuk mengotomatisasikan lebih dari 1 alat, penulis menambahkan code seperti di atas yang dimulai dari pin nomor 3 hingga pin nomor 13 dengan cara yang sama yaitu mendaftarkan pesan yang diwakilkan oleh 1 char huruf yang berfungsi mematikan/menghidupkan alat pada pin tersebut. Dan sebelumnya perlu didaftarkan terlebih dahulu OUTPUT pin yang diperlukan oleh board Arduino Uno pada bagian **void setup()**.

### **3.10 Android Application Berkomunikasi Dengan Mikrokontroler**

Pada bagian sebelumnya telah dibahas cara menghubungkan PC ke Mikrokontroler dan dari Mikrokontroler ke alat. Pada bagian ini dijelaskan lebih lanjut bagaimana cara mengkomunikasikan Mobile Device melalui Android Application ke Mikrokontroler melalui PC.

Langkah pertama yang dilakukan oleh penulis yaitu membuat sebuah file konfigurasi yang berekstensi PHP dimana file PHP tersebut nantinya akan digunakan untuk melakukan serial communication ke mikrokontroler melalui USB Serial yang terhubung dengan PC server dan board Arduino Uno dimana serial communication tersebut akan dilakukan ketika file PHP tersebut diakses oleh pengguna mobile device Android melalui Android Application yang dirancang. Sebelumnya, ditekankan terlebih dahulu untuk mengatur fitur **DMZ Setting** yang terletak pada router sehingga semua

paket yang akan dikirim akan dialihkan ke ip address PC Server letak file-file PHP yang dituju.

### 3.10.1 PHP Turn On/Off Device

Metode ini merupakan kunci dimana aplikasi android yang dirancang akan dapat berkomunikasi dengan mikrokontroler. Artinya, file PHP yang nantinya dibuat akan digunakan oleh aplikasi Android untuk melakukan serial communication ke board Arduino Uno.

Berikut contoh code pada salah satu file php yang dibuat oleh penulis dengan penamaan file yang kinerjanya sesuai dengan fungsi dari nama file tersebut yang bernama checkpin.php :

```
...
exec('mode COM7: baud=9600 data=8 stop=1 parity=n xon=no');
// COM7 = Lokasi PORT Arduino Uno.(nomor 7 tersebut menunjukkan nomor PORT
Arduino Uno)
// baud = bit rate per second, by default 9600
// data = data bits, by default = 8
// stop = stop bits, by default = 1
// parity = diisikan n sebagai perwakilan '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.";
}
else{
    mysql_query("update  devices  set  devicestatus=1
where devices.pin=".$pin."");
    fwrite($fp, chr(97)); // write 'a'
    // close serial communication
    fclose($fp);
    echo "device $pin on";
    break;
}
case "off":
    // serial communication for port COM7
    $fp = fopen ("COM7:", "w+");
    if (!$fp) {
        echo "Port cant be Opened.";
    }
    else{
        mysql_query("update  devices  set  devicestatus=0
where devices.pin=".$pin."");
        fwrite($fp, chr(98)); //write 'b'
        //close serial communication
        fclose($fp);
        echo "device $pin off";
        break;
    }
}
}

```

...

Tujuan pembuatan file php dengan nama file checkpin.php dengan code seperti di atas adalah untuk menghidupkan alat yang terletak pada pin 3(dikarenakan penulis memulai menggunakan digital input pada pin nomor 3 yang diasumsikan sebagai alat pertama) dengan mengakses file php di atas dimana fungsi dari code di dalam file php tersebut adalah memberikan input berupa char 'a' yang ASCII-nya adalah 97 dan char 'b' yang ASCII-nya adalah 98 ke board Arduino Uno melalui serial communication. Namun penulis melakukan hal tersebut sebanyak 22 kali yaitu dari 1 hingga 11 dengan masing-masing kondisi on dan off tersendiri pada setiap switch case-nya.

### 3.10.2 Android Activity POST

Pada bagian ini dijelaskan bagaimana cara membuat POST yang berfungsi untuk mengakses file PHP yang telah dibuat pada bagian sebelumnya dengan menggunakan metode HTTP POST seperti pada landasan teori yang digunakan oleh penulis.

Berikut contoh code yang dibuat oleh penulis yang menggunakan metode HTTP POST pada salah satu Android Activity:

```
public void postPHP(){
try{
    final String statusStr = "on";
    client = new DefaultHttpClient();

    post = new
    HttpPost("http://" + GlobalVar.ip + "/smarhome/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() {
                if(responseStr.equals("device "+pinStr+"
                on")){
                    Toast.makeText(getApplicationContext(),
                    responseStr, Toast.LENGTH_SHORT).show();

                button.setText("ON");

                }else{
                    Toast.makeText(getApplicationContext(),
                    "Connection Error",
                    Toast.LENGTH_SHORT).show();
                }
            }
        });
    } catch (Exception e) {}
}

```

Tujuan penulisan code di atas yaitu ketika method postPHP tersebut diakses oleh entitas salah satu Android Activity, misal button, maka Android Activity tersebut akan melakukan method HTTP POST untuk mengakses nama file PHP yang dituliskan di code dengan melempar parameter input yang sesuai dengan parameter file PHP tersebut dimana file PHP tersebut sendiri berfungsi untuk menghidupkan / mematikan alat yang ingin diinteraksikan.

### 3.11 Sistem Penjadwalan Alat Sistem Otomatisasi Rumah

Method ini dirancang oleh penulis khusus untuk menjadwalkan hidup/matinya alat listrik berdasarkan waktu yang telah diatur nantinya oleh pengguna aplikasi Android Sistem Otomatisasi Rumah.

### 3.11.1 PHP Turn On/Off Alat Tertentu

Untuk memudahkan sistem penjadwalan dari Sistem Otomatisasi Rumah yang akan dirancang oleh penulis, penulis membuat beberapa file PHP yang akan diberikan penamaan file yang berbeda-beda sesuai dengan fungsi dari namanya sendiri.

Berikut contoh file php yang diberinama onpin1.php :

```
...
exec('mode COM7: baud=9600 data=8 stop=1 parity=n xon=no');
// COM7 = Lokasi PORT Arduino Uno.(nomor 7 tersebut menunjukkan nomor PORT
Arduino Uno)
// baud = bit rate per second, by default 9600
// data = data bits, by default = 8
// stop = stop bits, by default = 1
//parity = diisikan n sebagai perwakilan 'none'(default)
//xon = no by default
$fp = fopen ("COM7:", "w+");// serial communication for port COM7
if (!$fp) {
    echo "Port cant be Opened.";
}
else{
    fwrite($fp, chr(97)); // wirte 'a'
    fclose($fp); // close serial communication
    echo "device on";
}
...

```

Tujuan pembuatan file php seperti di atas adalah ketika file php tersebut diakses oleh aplikasi desktop nantinya, LED yang berada pada pin nomor 3(pin

pertama yang digunakan) akan hidup. Penulis membuat file-file ini sebanyak 11 file dikali 2 dimana untuk setiap pinnya akan terdapat nama file onpin(x).php dan offpin(x).php dimana nilai x merupakan nomor pin yang akan diinteraksikan.

### 3.11.2 Timer Java Application

Method ini digunakan oleh penulis sebagai salah satu solusi untuk melakukan penjadwalan alat listrik yang didaftarkan pada Sistem Otomatisasi Rumah. Fungsi utamanya adalah untuk memeriksa waktu yang tepat untuk melakukan sebuah tindakan yang berupa mematikan / menhidupkan alat listrik yang terhubung pada board Arduino Uno melalui serial Communication yang dilakukan dengan mengakses file PHP yang telah dibuat pada bagian sebelumnya.

Contoh penggunaan method ini di aplikasi desktop Sistem Otomatisasi Rumah tertuang pada contoh code berikut :

```
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();

        date = now.get(Calendar.DATE);

        month = now.get(Calendar.MONTH);

        year = now.get(Calendar.YEAR);

        hour = now.get(Calendar.HOUR_OF_DAY);

        minute = now.get(Calendar.MINUTE);

        second = now.get(Calendar.SECOND);
```

```

getNameOfMonth(); //method untuk mengambil nama bulan

    date1 = new String(date+"-"+ monthString+"-" + year +", "+hour+":
    "+minute+": "+second);

    sysTime = new String(hour+": "+minute+": "+second);

    clock.setText(date1);

    newMonth(); //check bulan baru

    generatePrice(); //update harga

    checkSchedule(); //periksa jadwal

    refreshTable(); //refresh table
}
}

```

Tujuan dari penulisan code di atas adalah untuk mendaftarkan sebuah timer baru bernama ClockTickAction yang dipanggil melalui method setTime() dimana method setTime ini akan dipanggil ketika aplikasi desktop di run pertama kali.

Lalu untuk method yang akan dibahas oleh penulis untuk menjawab rumusan masalah nomor empat yaitu checkSchedule memiliki code seperti ini:

```

public void checkSchedule(){
    int id=1;
    for(int i=0; i<counter-1; i++){
        scheduleVect = null;

        // vector scheduleVect diisikan datanya melalui database

        scheduleVect = new ScheduleJDBCDAO().read("select * from schedule where
        devices_pin="+id);

        // mengisi ScheduleDom dengan vector scheduleVect

        ScheduleDom sd = (ScheduleDom)scheduleVect.elementAt(0);
    }
}

```

```

        //membandingkan waktu on/off dengan waktu saat ini
        if(java.sql.Time.valueOf(sysTime).equals(sd.getTimeOn())){

            turnOn(id);

        }else if( java.sql.Time.valueOf(sysTime).equals(sd.getTimeOff())){

            turnOff(id);

        }

        id++;

    }

}

```

Tujuan penulisan code di atas yaitu untuk melakukan pemeriksaan terhadap jadwal timeOn dan timeOff di database setiap alat. Lalu untuk method turnOn(id) atau turnoff(id) memiliki struktur code sebagai berikut :

```

public void turnOn(int id){

    DevicesJDBCDAO d = new DevicesJDBCDAO();
    deviceVect = null;

    //mengisi vector deviceVect dari database
    deviceVect = d.read("select * from devices where pin="+id);

    //mengisi domain DeviceDom dengan vector deviceVect
    DevicesDom dd = (DevicesDom)deviceVect.elementAt(0);
    int status = dd.getDeviceStatus(); //ambil nilai status dari dd

    if(status==0){

        //update database

        d.Edit("update devices set devicestatus = 1 where pin="+id);
        alimVersion.setValueAt("On", id-1, 2); // update tabel

        System.out.println("device "+dd.getName()+" on the "+dd.getRoom()+"
            is turned on");

    }

    try{

        //open file PHP yang berisi fungsi serial communication

```

```

URL url = new URL( "http://localhost/smarthome/onpin"+id+".php" );
URLConnection conn = (URLConnection) url.openConnection();
if( conn.getResponseCode() == HttpURLConnection.HTTP_OK ){
    InputStream is = conn.getInputStream(); // open file PHP
    // do something with the data here
}else{
    InputStream err = conn.getErrorStream(); // return error
    // err may have useful information.. but could be null see javadocs for
    more information
}
}catch(Exception e){
    System.out.println("error"); //jika file PHP tidak terbuka
}
}else if(status==1){
    System.out.println("device "+dd.getName()+" on the "+dd.getRoom()+" is
already turned on"); // jikalau sudah dalam kondisi nyala
}
}
}

```

Tujuan penulisan code di atas adalah untuk menghidupkan alat listrik pada id yang sesuai dengan parameter id yang dilempar oleh method checkSchedule().

### 3.11.3 Scheduling on Android Application

Pada salah satu Android Activity yang diberi nama Setting Screen Activity oleh penulis, nantinya akan terdapat sebuah HTTP POST yang berfungsi untuk mengakses file PHP yang telah dibuat di bagian sebelumnya.

Contoh code dari Activity ini akan dilampirkan pada bagian lampiran yang tujuan utamanya adalah mengatur jadwal timeOn dan timeOff dari setiap device pada Sistem Otomatisasi Rumah.

