



**PENGEMBANGAN APLIKASI “OKE GOOGLE” SEBAGAI KONTROL  
PENERANGAN RUMAH BERBASIS ARDUINO UNO**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh  
Gelara Sarjana Komputer pada Fakultas Sains dan Teknologi  
Universitas Pembangunan Panca Budi  
Medan

---

**SKRIPSI**

---

**OLEH**

**NAMA : ANGGA HANDACARDONIS**  
**NPM : 1514370240**  
**PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS PEMBANGUNAN PANCA BUDI  
MEDAN  
2019**

## ABSTRAK

### ANGGA HANDACARDONI S

## PENGEMBANGAN APLIKASI “*OKE GOOGLE*” SEBAGAI KONTROL PENERANGAN RUMAH BERBASIS ARDUINO UNO

2019

*Faktor usia sangat mempengaruhi kemampuan manusia dalam menjalankan aktivitas. Untuk melakukan hal-hal sederhana, terkadang sulit dilakukan sebagian orang. Kesulitan ini ditemui pada lansia dan anak-anak saat menjangkau saklar untuk menghidupkan peralatan elektronik seperti lampu. Pada tugas akhir ini dibuat sebuah alat pengendali lampu menggunakan perintah suara. Kemudahan yang disediakan adalah mengendalikan lampu dengan perintah suara menggunakan smartphone android. Alat ini dibuat untuk menggantikan fungsi saklar, sehingga dapat meng- ON/ OFF kan lampu tanpa harus menjangkau saklar yang jauh dengan user.*

*Adapun prinsip kerjanya dijelaskan berikut ini. User akan memberikan perintah suara melalui aplikasi Boarduino pada Handphone Android, dimana Android telah terhubung dengan Arduino Uno melalui Bluetooth HC-05 sebagai media transmisinya. Suara dideteksi oleh google voice kemudian disimpan dan direkam oleh microphone pada handphone. Suara yang direkam dikirimkan ke server google melalui internet, keudian google melakukan pembacaan dan memproses suara manusia. Dengan cara melakukan sampling pada suara yang sebelumnya berupa sinyal analog menjadi bit-bit biner (digital). Selanjutnya bit-bit tersebut diproses di dalam database google dan diterjemahkan dalam bentuk text. Setiap perintah suara akan diproses oleh Arduino Uno, jika perintah suara sesuai dengan program maka relay aktif sehingga lampu dapat dikendalikan.*

*Alat ini diharapkan mampu memenuhi kebutuhan dan tepat sasaran di masyarakat. Dan menjadi alat pemenuh kebutuhan, terutama untuk lansia dan anak-anak.*

**Kata Kunci:** *Handphone Android, Pengendali Lampu, Boarduino, Arduino Uno,*

*Bluetooth HC-05*

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR GAMBAR.....</b>	<b>vii</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
<b>BAB II LANDASAN TEORI</b>	
2.1 Suara.....	5
2.2 Pengolahan Sinyal Suara.....	4
2.3 Teknologi “OKE GOOGLE”.....	6
2.3.1 <i>Hidden Markov Models</i> .....	7
2.3.2 <i>Dinamic Time Warping</i> .....	8
2.3.3 <i>Neural Network</i> .....	8
2.4 Mikrokontroler .....	9
2.5 Arduino Uno .....	9
2.5.1 Daya (Power) .....	11
2.5.2 Input & Output .....	12
2.5.3 Komunikasi Arduino.....	13
2.5.4 Software Arduino .....	14
2.5.5 Bahasa Pemrograman Arduino Berbasis Bahasa C .....	15
2.6 Bluetooth HC-05 .....	18
2.7 Relay .....	20
2.8 HandPhone Android.....	21
2.8.1 Kelebihan Android.....	22
2.8.2 Kekurangan Android.....	23
2.9 UML (Unified Modeling Language).....	23
2.9.1 Use Case Diagram.....	24
2.9.2 Diagram Aktivitas (Activity Diagram) .....	25
2.9.3 Diagram Urutan (Sequence Diagram).....	26
2.10 FlowChart .....	27
2.11 Program App Inventor Pada Pemrograman Android.....	28
<b>BAB III METODE PENELITIAN</b>	
3.1 Tahapan Penelitian .....	33
3.2 Metode Pengumpulan Data .....	34

3.3	Analisis Sistem yang sedang berjalan .....	35
3.3.1	FlowChart Sistem yang sedang nerjalan .....	35
3.3.2	Analisis Sistem yang ditawarkan .....	36
3.3.3	Flowchart Sistem yang ditawarkan .....	36
3.4	Rancangan Penelitian .....	38
3.4.1	Perancangan Perangkat Keras ( <i>HardWare</i> ) .....	38
3.4.1.1	Diagram Blok Sistem .....	38
3.4.1.2	Rangkaian Bluetooth HC-05 .....	39
3.4.1.3	Rangkaian Modul Relay .....	40
3.4.1.4	Rangkaian Keseluruhan .....	41
3.4.2	Perancangan Perangkat Lunak ( <i>SoftWare</i> ).....	41
3.4.2.1	Use Case Diagram.....	42
3.4.2.2	Activity Diagram.....	43
3.4.2.3	FlowChart Hardware.....	44
3.4.2.4	FlowChart Android .....	45
3.4.3	Lokasi Perancangan .....	48
3.4.4	Langkah Pengujian Alat.....	48

#### **BAB IV HASIL DAN PEMBAHASAN**

4.1	Kebutuhan Spesifikasi Minimum Hardware dan Software.....	50
4.1.1	Kebutuhan Spesifikasi Minimum Hardware.....	50
4.1.2	Kebutuhan Spesifikasi Minimum Software .....	51
4.2	Pengujian Aplikasi dan Pembahasan .....	51
4.2.1	Pengujian Arduino .....	51
4.2.2	Pengujian Aplikasi Android.....	53
4.2.3	Pengujian Relay .....	54
4.2.4	Pengujian Bluetooth.....	55
4.2.5	Pengujian Seluruh Sistem .....	56
4.2.6	Tampilan Interface .....	56
4.2.7	Pengujian Perancangan Alat Kendali Lampu .....	58

#### **BAB V PENUTUP**

5.1	Simpulan .....	67
5.2	Saran.....	67

#### **DAFTAR PUSTAKA**

#### **BIOGRAFI PENULIS**

#### **LAMPIRAN-LAMPIRAN**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi dan informasi yang berkembang pesat sangat berperan penting dalam kehidupan sehari-hari. Kemudahan yang ditawarkan dalam melakukan pekerjaan sehari-hari dapat meringankan pekerjaan-pekerjaan manusia. Banyak alat-alat diciptakan untuk memenuhi kebutuhan dan kepuasan masyarakat agar pekerjaan dapat dikerjakan lebih cepat dan efisien. Sistem otomatisasi peralatan elektronik merupakan salah satu kemudahan yang disediakan untuk mengendalikan peralatan listrik dirumah, seperti menghidupkan dan mematikan lampu pada ruangan. Saklar lampu yang sulit dijangkau oleh lansia dan anak-anak cukup menyulitkan mereka saat ingin menghidupkan dan mematikan lampu ruangan di dalam rumah. Untuk itu, pembaharuan otomasi alat-alat dengan memanfaatkan teknologi sangat dibutuhkan untuk membantu dalam sistem otomatisasi saklar lampu.

Beberapa penelitian sebelumnya yang terkait dengan sistem otomatisasi pada saklar lampu telah dilakukan oleh Budi Novianto *et al*, 2015. Dalam penelitiannya, peneliti merancang sistem kendali dan monitoring lampu berbasis *Short Message Service* (SMS). Dalam penelitian ini, mereka merancang sistem kendali dan monitoring pada lampu berbasis *Short Message Service* (SMS) yang dapat mengendalikan lampu secara jarak jauh dan mengecek kondisi lampu. Adapun sebagai umpan balik pada peralatan ini menggunakan LDR sebagai

sensor, GSM *shield* 900 sebagai penghubung informasi ke handphone dan Arduino uno sebagai pusat pengendali dan pengolahan data yang nantinya akan memberikan perintah kepada *relay* dan akan di teruskan untuk meyalakan dan mematikan lampu. Penelitian yang sama dilakukan oleh Trisiani Dewi Hendrawati dan Indra Lesmana pada tahun 2016. Dalam penelitiannya, peneliti merancang aplikasi smart home dengan merakit saklar lampu otomatis dan monitoring suhu. Peneliti merancang sistem dengan menggunakan modul mikrokontroler Arduino Uno dan VB.

Berdasarkan penelitian tersebut penulis ingin melanjutkan pengembangan alat otomatisasi saklar lampudengan menggunakan perintah suara sebagai media menghidupkan dan mematikan saklar lampu. Para lansia dan anak-anak dapat mematikan lampu yang berada di ruangan kita tanpa harus menjangkau saklar. Dengan *activationtooth* melalui aplikasi. Android merupakan solusi yang tepat dan efisien, sehingga kita dapat dengan mudah mengontrol alat-alat yang ingin digunakan dengan perintah suara tanpa harus mencari saklar yang berada ditempat yang berbeda-beda. Melihat kebutuhan masyarakat yang semakin meningkat dan alat pemenuhan yang terus dicari, maka penulis ingin membuat sebuah alat yang dapat menghidupkan dan mematikan lampu ruangan menggunakan perintah suara dengan judul Tugas Akhir **“Pengembangan Aplikasi “Oke Google” Sebagai Kontrol Penerangan Rumah Berbasis Arduino Uno”**.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka rumusan masalah adalah :

1. Bagaimana merancang pengendali lampu dengan perintah suara menggunakan android via *bluetooth* sebagai media transmisinya.
2. Bagaimana membuat pengendali lampu dengan perintah suara menggunakan android via *bluetooth* sebagai media transmisinya.

## 1.3 Batasan Masalah

Permasalahan pada proyek Penelitian ini dibatasi pada:

1. Menggunakan mikrokontroler ATmega 328.
2. Pembahasan tentang beberapa komponen seperti pemanfaatan *smartphone* android sebagai control lampu ruangan yang terkoneksi oleh *Google* sebagai media pembaca perintah.
3. Penyalaan lampu menggunakan perintah suara.

## 1.4 Tujuan Penelitian

Ada pun tujuan dari Penelitian ini adalah :

1. Membuat suatu alat yang mampu mengendalikan lampu menggunakan perintah suara.
2. Menggantikan fungsi saklar untuk menghidupkan dan mematikan lampu menggunakan perintah suara.

## **1.5 Manfaat Penelitian**

Adapun manfaat penulisan Penelitian ini adalah:

1. Memberikan pengetahuan baru bagi penulis dalam membuat implementasi Arduino uno sebagai pembelajaran sistem kendali untuk mengontrol lampu ruangan dengan perintah suara.
2. Dapat dimanfaatkan oleh universitas sebagai bahan pembelajaran dan praktikum pada pembahasan sistem kontrol.
3. Meringankan dan memudahkan pekerjaan, dapat mengendalikan lampu tanpa harus mencari saklar.
4. Menggantikan fungsi saklar.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Suara**

Suara adalah sinyal atau gelombang yang merambat dengan frekuensi dan amplitude tertentu melalui media perantara yang dihantarkannya seperti media air, udara maupun bendapadat. Manusia dapat berkomunikasi dengan manusia lainnya dengan suara. Pembangkitan ucapan manusia dimulai dengan awal konsep dari gagasan yang ingin disampaikan pada pendengar. Pengucap mengubah gagasan tadi dalam struktur *linguistic* dengan memilih kata atau frasa yang secara tepat dapat mewakili dan membawakannya dengan tata bahasa yang dimengerti antara pengucap dan pendengar. Ucapan yang diucapkan memiliki tujuan tertentu dengan asumsi bahwa ucapan tersebut diucapkan secara benar, dapat diterima, dan dipahami oleh pendengar yang dituju. Pembangkitan ucapan pada hakekatnya berhubungan dengan kemampuan mendengar. Sinyal ucapan dibangkitkan oleh organ *vocal* dan ditransmisikan melalui udara menuju telinga pendengar. Frekuensi fundamental suara laki-laki lebih rendah dibandingkan frekuensi fundamental suara perempuan yaitu  $f_0$  perempuan diatas 200 Hz sedangkan  $f_0$  laki-laki dibawah 170 Hz. (Rolly & Hakiem, 2015)

#### **2.2 Pengolahan Sinyal Suara**

Definisi dari sinyal suara yaitu suatu sinyal yang mewakili dari suara. Sinyal suara dibentuk dari kombinasi berbagai frekuensi pada berbagai amplitudo

dan fasa. Pengolahan suara adalah suatu perkembangan teknik dan sistem yang memungkinkan komputer suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat. Kata-kata diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasi kata-kata tersebut tersebut, hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan atau dapat dibaca oleh perangkat teknologi sebagai sebuah komando untuk melakukan suatu pekerjaan. (Ishanto & Hidayat, 2014)

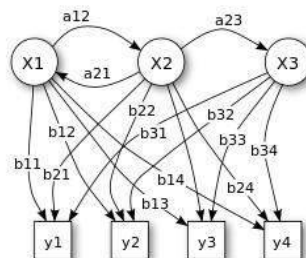
### **2.3 Teknologi “OKE GOOGLE”**

*Voice Command Recognition System* atau yang sering kali disebut dengan teknologi *Speech Recognition* (pengenalan kalimat atau kata) dalam ilmu komputer dan teknik elektronik adalah sebuah sistem yang mengubah kalimat suara menjadi kode - kode digital yang berfungsi sebagai perintah untuk melakukan sesuatu pada sistem, sebagai contoh adalah mengemudikan kendaraan, mematikan atau menghidupkan lampu, maupun tugas – tugas yang lainnya. Beberapa *sistem speech recognition* biasanya menggunakan *speaker independent speech recognition* sementara yang lainnya menggunakan *Training*. *Training* ini adalah pelatihan yang dilakukan oleh *user* terhadap *system Speech Recognition* dimana seorang *user* akan membacakan teks-teks tertentu yang kemudian secara otomatis akan dimasukan kedalam sistem *Speech Recognition*.

Ada 3 buah algoritma yang digunakan oleh *speech recognition* pada masa sekarang ini yaitu dengan *Hidden Markov Models*, *Dynamic Time Warping*, dan *Neural Network*. Berikut dibawah ini adalah penjelasan mengenai algoritma-algoritma tersebut :

### 2.3.1 *Hidden Markov Models*

Algoritma yang digunakan pada sistem *speech recognition* adalah algoritma *Hidden Markov Model*. Algoritma ini menggunakan permodelan statistik yang menghasilkan keluaran berupa susunan simbol atau jumlah. HMM digunakan pada sistem ini disebabkan karena kalimat dapat dilihat sebagai *piecewise stationary signal*, sehingga setiap perkataan dapat dilihat sebagai pendekatan sebuah proses yang tidak bergerak/tetap.



**Gambar 2.1** Algoritma Hidden Markov

Sumber: Satria, Yanti & Maulinda (2017)

Model Algoritma ini akan menghasilkan sebuah urutan dari *vector real-valued* dengan  $n$ -dimensi. Vektor ini akan memiliki *koefisien kepstral* (adalah hasil dari bentuk *Inverse Fourier transform* pada logaritma yang mencari *spectrum* dari sebuah signal), yang didapat dari sebuah transformasi *fourier* dengan waktu yang pendek dari perkataan dan memotong-motong *spectrum* tersebut dengan transformasi *Cosine* dan mengambil *koefisien* yang terbesar. Algoritma ini akan memiliki setiap bentuk sebuah *statistic* distribusi yang

merupakan campuran dari *Diagonal Covariance Gaussian*, sehingga akan memberikan kemungkinan untuk setiap *vector* yang teramati. Setiap kata atau kalimat akan menghasilkan distribusi *statistic hidden markov model*, sehingga dapat digunakan untuk memperkirakan kalimat yang diucapkan oleh pengguna.

### **2.3.2 *Dynamic Time Warping***

*Dynamic Time Warping* adalah sebuah algoritma untuk mengukur persamaan antara 2 buah urutan yang dapat berbeda didalam waktu maupun kecepatan. Sebagai contoh adalah seseorang berjalan lambat maupun cepat maka data tersebut dapat di jadikan representasi linear sehingga dapat di analisa dengan DTW. Dalam bahasa umum, algoritma ini adalah sebuah metode untuk menemukan persamaan yang optimal antara 2 urutan dengan pembatasan tertentu.

### **2.3.3 *Neural Network***

Algoritma ini digunakan untuk memperkirakan kemungkinan dari sebuah kata-kata. Dengan percobaan yang membedakan antara yang satu dengan yang lainnya sehingga sangat efisien penggunaannya. Pada masa kini algoritma RNN (*Recurrent Neural Network*) dan TDNN (*Time delay Neural Network*) telah digunakan untuk menentukan kekurangan sementara yang tersembunyi pada pengenalan kata dan menggunakan informasi yang tersedia untuk menghasilkan jalan yang paling efisien dan paling efektif pada pengenalan kalimat. Namun hal ini menghasilkan biaya komputasi yang tinggi (besarnya komputasi akan menghasilkan kecepatan komputasi yang rendah) sehingga tidak efektif dalam

pengenalan kalimat. Pada masa kini penelitian masih tetap dilakukan sehingga memastikan bahwa TDNN dan RNN dapat digunakan namun dengan menekan nilai komputasi yang rendah.

#### **2.4 Mikrokontroller**

Mikrokontroller adalah suatu chip cerdas yang dapat digunakan sebagai pengontrol utama sistem elektronika, misalnya sistem pengukur suhu digital (*thermometer digital*), sistem keamanan rumah, sistem kendali mesin industri, robot penjinak bom, dan lain-lain. Hal ini dikarenakan didalam chip tersebut sudah ada unit pemroses, memori ROM (*Read Only Memory*), RAM (*Random Access Memory*), I/O, dan fasilitas pendukung lainnya.

Mikrokontroller adalah sistem mikroprosesor lengkap yang terkandung di dalam sebuah chip. Mikrokontroller berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena di dalam sebuah mikrokontroller umumnya juga telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O, sedangkan di dalam mikroprosesor umumnya hanya berisi CPU saja (Ikhsan & Kurniawa, 2015)

#### **2.5 Arduino UNO**

Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang didalamnya terdapat komponen utama yaitu sebuah chip mikrokontroller dengan jenis AVR tujuan dalam keadaan aktif dan menerima SMS yang dikirim, maka akan ada konfirmasi program sedangkan RAM untuk menyimpan data

sementara dari perusahaan Atmel. Mikrokontroler itu sendiri adalah chip atau IC (*Integrated Circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input tersebut dan kemudian menghasilkan output sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai “otak” yang mengendalikan input, proses dan output sebuah rangkaian elektronik. (Syahminan, 2017)

Arduino Uno adalah sebuah *board* mikrokontroler yang didasarkan pada Atmega328. Arduino UNO mempunyai 14 pin digital *input/output* (6 diantaranya dapat digunakan sebagai luaran PWM), 6 masukan analog, sebuah osilator 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Arduino UNO mampu *men-support* mikrokontroler, dan dapat dikoneksikan dengan komputer menggunakan kabel USB. (Satria, Yanti & Maulinda, 2017)



**Gambar 2.2** Board Arduino Uno  
Sumber: Satria, Yanti & Maulinda (2017)

**Tabel 2.1** Tabel Deskripsi Arduino Uno

Mikrokontroler	ATmega 328
Tegangan Pengoperasian	5V

**Tabel 2.1** Tabel Deskripsi Arduino Uno (Lanjutan)

Tegangan Input yang disarankan	7-12 V
Batas Tegangan Input	6-20 V
Jumlah pin I/O digital	14 pin digital ( 6 diantaranya menyediakan keluaran PWM)
Jumlah pin input Analog	6 pin
Arus DC tiap pin I/O	40mA
Arus DC untuk pin 3,3 V	50mA
<i>Memori Flash</i>	32 KB (ATmega 328) sekitar 0,5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega 328)
EPROM	1 KB (ATmega 328)
<i>Clock Speed</i>	16 MHz

Sumber: Wicaksono & Hidayat, (2017)

### 2.5.1 Daya (*Power*)

Arduino dapat diberikan *power* melalui koneksi USB atau *power supply*. *Power*-nya diseleksi secara otomatis. *Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan mencolok *jack* adaptor pada koneksi port input *supply*. *Board* arduino dapat dioperasikan menggunakan *supply* dari luar sebesar 6 - 20 volt. Jika *supply* kurang dari 7V, kadangkala pin 5V akan menyuplai kurang dari 5 volt dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 V, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 sampai 12 volt.

Penjelasan pada pin power adalah sebagai berikut :

#### 1. Vin

Tegangan *input* ke *board* arduino ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 volt dari koneksi USB atau tegangan yang

diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan pin ini.

#### 2. 5V

Regulasi *power supply* digunakan untuk *power* mikrokontroller dan komponen lainnya pada *board*. 5V dapat melalui  $V_{in}$  menggunakan regulator pada *board*, atau *supply* oleh USB atau *supply* regulasi 5V lainnya.

#### 3. 3V3

Suplai 3.3 volt didapat oleh FTDI chip yang ada di *board*. Arus masimumnya adalah 50mA

#### 4. Pin *Ground*

Berfungsi sebagai jalur *ground* pada Arduino.

#### 5. Memori

ATmega328 memiliki 32 KB flash memori untuk menyimpan kode, juga 2 KB yang digunakan untuk *bootloader*. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

### 2.5.2 Input & Output

Setiap 14 pin *digital* pada arduino dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. *Input* atau *output* dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau



menerima *maximum* 40 mA dan memiliki *internal pull-up resistor (disconnected oleh default)* 20-50K Ohm.

Beberapa pin memiliki fungsi sebagai berikut :

1. Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL *data serial*. Pin ini terhubung pada pin yang koresponding dari USB ke TTL chip serial.
2. *Interupt eksternal* : 2 dan 3. Pin ini dapat dikonfigurasi untuk *trigger* sebuah *interap* pada *low value*, *rising* atau *falling edge*, atau perubahan nilai.
3. PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit output PWM dengan fungsi *analogWrite()*.
4. SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini *men-support* komunikasi SPI, yang mana masih mendukung *hardware*, yang tidak termasuk pada bahasa arduino.
5. LED : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup, ketika pin LOW, LED mati.

### 2.5.3 Komunikasi Arduino

Uno Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin *digital* 0 (RX) dan 1 (TX). *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Ini diperlukan.

Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer.

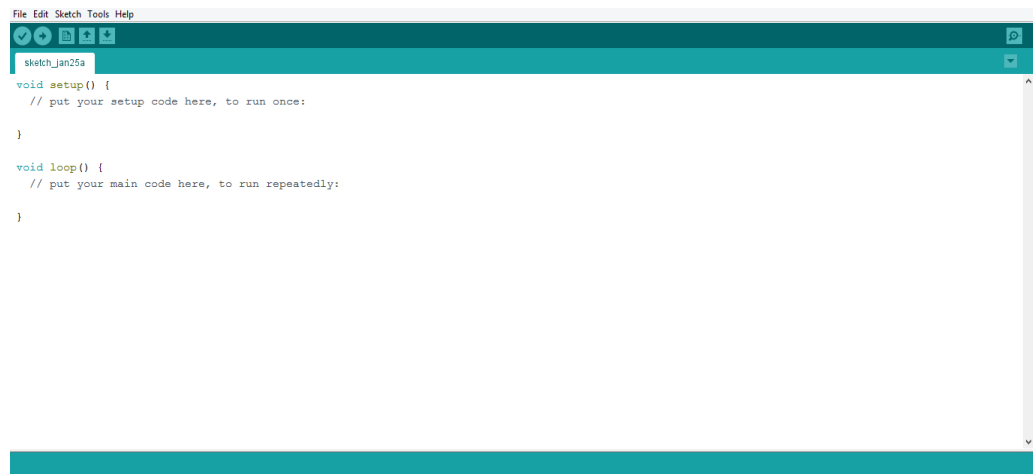
#### 2.5.4 Software Arduino

Arduino Uno dapat diprogram dengan perangkat lunak Arduino . Pada ATmega328 di Arduino terdapat *bootloader* yang memungkinkan Anda untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware eksternal*.

IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino.

Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata “*sketch*” digunakan secara bergantian dengan “kode program” dimana keduanya memiliki arti yang sama.



**Gambar 2.3** Tampilan IDE Arduino

Sumber: Hari Santoso, (2015)

### 2.5.5 Bahasa Pemrograman Arduino Berbasis Bahasa C

Seperti yang telah dijelaskan diatas program Arduino sendiri menggunakan bahasa C. Banyak sekali terdapat bahasa pemrograman tingkat tinggi (*high level language*) seperti *pascal*, *basic*, *cobol*, dan lainnya. Walaupun demikian, sebagian besar dari para programmer profesional masih tetap memilih bahasa C sebagai bahasa yang lebih unggul, berikut alasan-alasannya:

1. Bahasa C merupakan bahasa yang *powerful* dan *fleksibel* yang telah terbukti dapat menyelesaikan program-program besar seperti pembuatan sistem operasi, pengolah gambar (seperti pembuatan game) dan juga pembuatan kompilator bahasa pemrograman baru.

2. Bahasa C merupakan bahasa yang *portabel* sehingga dapat dijalankan di beberapa sistem operasi yang berbeda. Sebagai contoh program yang kita tulis dalam sistem operasi windows dapat kita kompilasi didalam sistem operasi linux dengan sedikit atau pun tanpa perubahan sama sekali.
3. Bahasa C merupakan bahasa yang sangat populer dan banyak digunakan oleh programer berpengalaman sehingga kemungkinan besar *library* pemrograman telah banyak disediakan oleh pihak luar/lain dan dapat diperoleh dengan mudah.
4. Bahasa C merupakan bahasa yang bersifat modular, yaitu tersusun atas rutin-rutin tertentu yang dinamakan dengan fungsi (*function*) dan fungsi-fungsi tersebut dapat digunakan kembali untuk pembuatan program-program lainnya tanpa harus menulis ulang implementasinya.
5. Bahasa C merupakan bahasa tingkat menengah (*middle level language*) sehingga mudah untuk melakukan interface (pembuatan program antar muka) ke perangkat keras.
6. Struktur penulisan program dalam bahasa C harus memiliki fungsi utama, yang bernama `main()`. Fungsi inilah yang akan dipanggil pertama kali pada saat proses eksekusi program. Artinya apabila kita mempunyai fungsi lain selain fungsi utama, maka fungsi lain tersebut baru akan dipanggil pada saat digunakan.

Oleh karena itu bahasa C merupakan bahasa prosedural yang menerapkan konsep runtutan (program dieksekusi per baris dari atas ke bawah secara berurutan), maka apabila kita menuliskan fungsi-fungsi lain tersebut dibawah

fungsi utama, maka kita harus menuliskan bagian *prototype*, hal ini dimaksudkan untuk mengenalkan terlebih dahulu kepada kompiler daftar fungsi yang akan digunakan di dalam program. Namun apabila kita menuliskan fungsi-fungsi lain tersebut diatas atau sebelum fungsi utama, maka kita tidak perlu lagi untuk menuliskan bagian *prototype* diatas. ( Ihsanto & Hidayat,2014 )

Selain itu juga dalam bahasa C kita akan mengenal *file header*, biasa ditulis dengan ekstensi h(\*.h), adalah file bantuan yang yang digunakan untuk menyimpan daftar-daftar fungsi yang akan digunakan dalam program. Bagi anda yang sebelumnya pernah mempelajari bahasa pascal, *file header* ini serupa dengan unit. Dalam bahasa C, file header standar yang untuk proses *input/output* adalah *<stdio.h>*.

Perlu sekali untuk diperhatikan bahwa apabila kita menggunakan *file header* yang telah disediakan oleh kompilator, maka kita harus menuliskannya didalam tanda ‘<’ dan ‘>’ (misalnya *<stdio.h>*). Namun apabila menggunakan *file header* yang kita buat sendiri, maka file tersebut ditulis diantara tanda “ dan ” (misalnya “coba *header.h*”). perbedaan antara keduanya terletak pada saat pencarian file tersebut. Apabila kita menggunakan tanda <>, maka file tersebut dianggap berada pada direktori default yang telah ditentukan oleh kompilator. Sedangkan apabila kita menggunakan tanda “”, maka *file header* dapat kita dapat tentukan sendiri lokasinya.

*File header* yang akan kita gunakan harus kita daftarkan dengan menggunakan directive *#include*. Directive *#include* ini berfungsi untuk memberi

tahu kepada kompilator bahwa program yang kita buat akan menggunakan file-file yang didaftarkan. Berikut ini contoh penggunaan directive *#include*.

```
#include<stdio.h>  
  
#include<stdlib.h>  
  
#include"myheader.h"
```

Setiap kita akan menggunakan fungsi tertentu yang disimpan dalam sebuah *file header*, maka kita juga harus mendaftarkan *file header*nya dengan menggunakan directive *#include*. Sebagai contoh, kita akan menggunakan fungsi *getch()* dalam program, maka kita harus mendaftarkan *file header*<*conio.h*>.

## 2.6 **Bluetooth HC-05**

*Bluetooth* adalah protokol komunikasi *wireless* yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain. Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe HC-05. modul *Bluetooth* HC-05 merupakan salah satu modul *Bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *Bluetooth* HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda - beda. (Dian Artanto, 2015)

*Bluetooth* merupakan sebuah teknologi komunikasi *wireless* yang beroperasi pada pita frekuensi 2,4 – 2,83 GHz *unlicensed ISM (Industrial, Scientific and Medical)*. Bluetooth mampu menyediakan layanan komunikasi data

antara *host-host* bluetooth dengan jarak jangkauan layanan yang terbatas. (Silvia, Haritman, Muladi, 2014)

Ada dua jenis *bluetooth* ke modul serial dengan ganjil dan genap. *Bluetooth* seri bernomor ganjil sebagai HC-05 atau HC-03 adalah versi perbaikan dari *Bluetooth* untuk Serial Modul HC-06 atau HC-04. *Bluetooth* ke serial modul HC-05 dapat ditetapkan sebagai *master* atau *slave* perangkat seperti HC-06 modul yang hanya bisa digunakan sebagai *Slave*. *Bluetooth* konfigurasi modul pin Serial HC-05 ditunjukkan pada Gambar 2.4. (Akhmad Zainuri, 2015)



**Gambar 2.4** Module Bluetooth Hc-05  
Sumber : Akhmad Zainuri, (2015)

Deskripsi modul HC-05:

1. Level tegangan kerja 3.3V.
2. Modul memiliki 2 mode kerja (pemilihan mode dengan mengubah status pin 34 – KEY):
  - a. Auto-connect.
  - b. Mode ODAP, Anda dapat mengirim perintah AT untuk berkomunikasi dengan modul. Dengan mengubah status 34 kaki (KEY), Anda dapat mengkonfigurasi modus operasi modul:

- c. Untuk membuat modul dalam mode koneksi otomatis: KEY ke kondisi *floating* (tidak terhubung state).
  - d. Untuk modul bekerja di bawah modus respon perintah: KEY = '0' (koneksi *ground*) dan KEY = '1' (terhubung ke Vcc) sekarang dapat menggunakan perintah AT untuk berkomunikasi.
3. Baudrate 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, dapat di set sesuai dengan kebutuhan user.
  4. Kebutuhan Arus : Pairing 20~30mA. Setelah Pair: 8mA
  5. Frekuensi yang digunakan : 2.5 GHz . (Akhmad Zainuri, 2015)

## 2.7 Relay

*Relay* dapat digunakan untuk menyambung atau memutus tegangan dan arus yang lebih besar dibandingkan yang dapat dilakukan oleh Arduino. *Relay* menyediakan isolasi lengkap antara rangkaian tegangan rendah disisi Arduino dengan rangkaian tegangan tinggi yang akan digunakan. *Relay* membutuhkan tegangan 5 Vdc dan memiliki rentan arus sebesar 15-20mA. *relay* ini dapat mengontrol peralatan elektronik tegangan tinggi dibawah 250Vac dengan arus maksimal sebesar 10A. (Manoj Kumar et al., 2017).



**Gambar 2.5** Relay



Berikut spesifikasi Relay channel :

- |   |                                |
|---|--------------------------------|
| 1. <i>External Power Supply Voltage</i> | 7-12V (>200 mA)                |
| 2. <i>Relay Max Switching Voltage</i>   | 240V AC/ 60VDC                 |
| 3. <i>Relay Max Switching Current</i>   | 5A                             |
| 4. <i>Contact Load Capacity</i>         | 1A250VAC,3A120VAC/24VDC        |
| 5. <i>Number of Relays</i>              | 4pcs                           |
| 6. <i>Wireless Output Interface</i>     | (serial)xbee/Bluetooth Bee/WPM |

## **2.8 Handphone Android**

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. (Nazaruddin Safaat, 2015)

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, *konsorsium* dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler.

### 2.8.1 Kelebihan Android

Adapun kelebihan yang terdapat di android adalah sebagai berikut :

1. *Switching* dan *multitasking* yang lebih baik Android sangat mendukung *multitasking* aplikasi, kini hal tersebut kembali ditingkatkan. Dalam *Honeycomb* pengguna dapat dengan mudah berpindah aplikasi hanya dengan menyentuh sebuah *icon* pada *system bar*.
2. Kapasitas yang lebih baik untuk beragam *widget* Kapabilitas terhadap beragam *widget* dijanjikan bakal makin memanjakan para penggunanya. Contohnya *widget* untuk email Gmail yang dipamerkan Google, pengguna tidak perlu membuka aplikasi Gmail untuk melihat isi di dalamnya.
3. Peningkatan kemampuan *copy-paste* Beberapa seri Android terdahulu memang sudah bisa melakukan *copypaste*, namun beberapa pengguna masalah pemilihan teks yang agak sulit. Kini hal tersebut coba diselesaikan, selain *copypaste* Google juga menambah *share it* pada teks yang diseleksi.
4. *Browser Chrome* Lebih Cepat Ada satu fitur yang hilang dalam *browser Chrome* yang diletakkan pada Android terdahulu, kemampuan Tab. Chrome yang ada di *Honeycomb* kini dapat melakukan hal tersebut. Selain itu pengguna juga bisa mensinkronisasi antara browser di ponsel dengan Chrome yang ada di komputer.
5. *Notifikasi* yang Mudah Terdengar. Dengan layar yang lebih besar, otomatis membuat Google lebih leluasa menempatkan *notifikasi* pada layar.

6. Peningkatan *Drag and Drop* serta *Multitouch* Ukuran layar yang lebih besar, menuntut Google untuk meningkatkan kemampuan *multitouch* di dalam Android, tak terkecuali fitur *drag and drop*. Pada demo yang ditayangkan, pengguna bisa melakukan *drag and drop* untuk memindahkan email di dalam aplikasi *Gmail*.

### **2.8.2 Kekurangan Android**

Adapun kekurangan yang ada pada android adalah sebagai berikut :

1. Koneksi Internet yang terus menerus. Kebanyakan ponsel Android memerlukan koneksi internet yang simultan atau terus menerus aktif, itu artinya anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan dan baterai yang boros karena GPRS yang terus menyala.
2. Aplikasi di Ponsel Android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu ada Iklan yang terpampang. (Nazaruddin Safaat, 2015)

### **2.9 UML (*Unified Modeling Language*)**





Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language (UML)* adalah Bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Ade Hendini, 2016).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

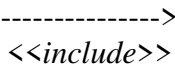
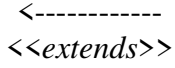
### 2.9.1 Use Case Diagram

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

**Tabel 2.2** Simbol Use Case Diagram

Simbol	Keterangan
	Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
	Actor atau Aktor adalah Abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan Use Case, tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.

**Tabel 2.2** Simbol Use Case Diagram (Lanjutan)


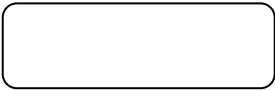
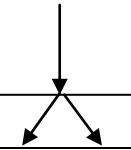
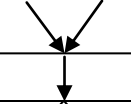


Simbol	Keterangan
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (required) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

Sumber : Ade Hendini, (2016)

### 2.9.2 Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

**Tabel 2.3** Simbol Diagram *Activity*

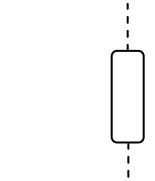
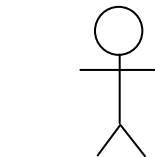

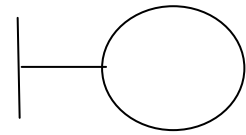
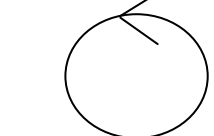
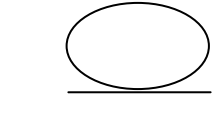
Simbol	Keterangan
	End Point , Akhir aktivitas
	<i>Activities</i> , Menggambarkan suatu proses
	Fork/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision point</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true of false</i>
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

Sumber : Ade Hendini, (2016)

### 2.9.3 Diagram Urutan (*Sequence Diagram*)

*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

**Tabel 2.4** Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi
2		<i>Actor</i>	Digunakan untuk menggambarkan user / pengguna
3		<i>Message</i>	Spesifikasi dari Komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
4		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form
5		<i>Control Class</i>	Digunakan untuk menghubungkan <i>boundary</i> dengan table
6		<i>Entity Class</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan di lakukan


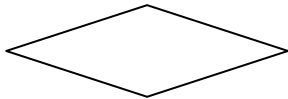
Sumber : Ade Hendini, (2016)

## 2.10 Flow Chart

*Flowchart* merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis. (Eka Iswandi, 2015)


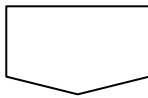



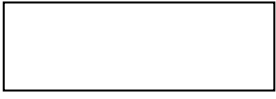

Menurut Adelia, Jimmy (2011:116) *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa mempersentasikan komponen – komponen dalam Bahasa pemrograman.

**Tabel 2.5** Simbol-simbol *Flowchart*

No.	Simbol	Keterangan
1.		Permulaan sub program
2.		Perbandingan, pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya

Sumber : Ade Hendini, (2016)

**Tabel 2.5** Simbol-simbol *Flowchart* (Lanjutan)

No.	Simbol	Keterangan
3.		Penghubung bagian-bagian flowchart yang berada pada satu halaman
4.		Penghubung bagian-bagian flowchart yang berada pada halaman berbeda
5.		Permulaan/akhir program
6.		Arah aliran program
7.		Proses inisialisasi/pemberian harga awal
8.		Proses penghitung/proses pengolahan data
9.		Proses <i>input/output</i> data

Sumber : Ade Hendini, (2016)

### 2.11 Program *App Inventor* Pada Pemrograman Android

Mulyadi (2013: 1) menjelaskan *App Inventor* adalah sebuah *tool* untuk membuat aplikasi android yang berbasis *visual block programming*, sehingga



pengguna bisa membuat aplikasi tanpa melakukan *coding*. *Visual block programming* maksudnya adalah dalam penggunaannya *user* akan melihat, menggunakan, menyusun dan *drag-drops* “blok” yang merupakan simbol-simbol perintah dan fungsi –*event handler* tertentu dalam membuat aplikasi, dan secara sederhana bisa disebut tanpa menuliskan kode program.

Aplikasi *App Inventor* ini pada dasarnya adalah aplikasi yang disediakan oleh google dan sekarang di-*maintenance* oleh *Massachusetts Institute of Technology* (MIT). Aplikasi ini selesai dibuat pada 12 juli 2010 dan dirilis untuk public pada 31 Desember 2011. *App Inventor* sekarang dipegang oleh MIT *Centre for Mobile Learning* dengan nama *MIT App Inventor*. Gambar berikut adalah gambar tampilan aplikasi *App Inventor*:

Dengan menggunakan *App Inventor* ini, ada beberapa aplikasi yang dapat dibuat diantaranya yaitu:

1. Aplikasi *game*
2. Aplikasi berbasis *tracking* lokasi
3. Aplikasi SMS
4. Aplikasi berbasis web
5. Aplikasi kompleks

Aplikasi *App Inventor* memiliki tiga opsi dalam proses pengujian sistem, diantaranya adalah:

1. Membuat aplikasi dengan perangkat Android dengan koneksi *Wi-Fi*. Bisa dilihat pada gambar 2.6 sebagai berikut.



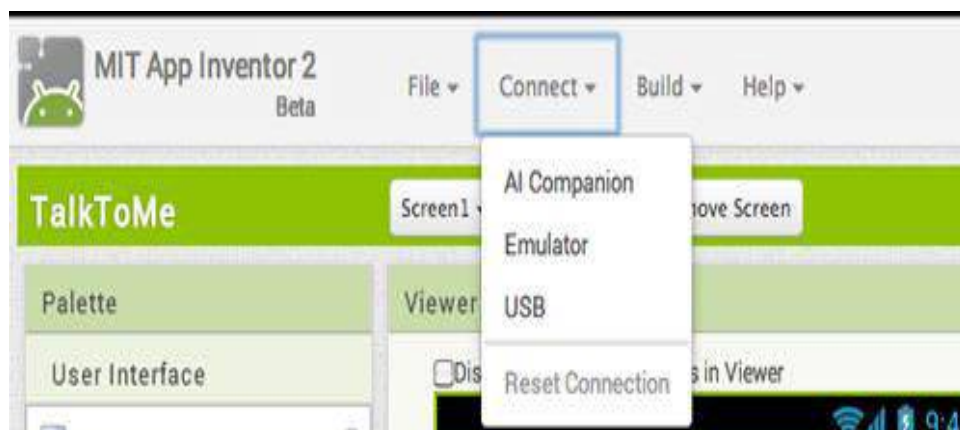
**Gambar 2.6** Koneksi Android dengan *Wi-Fi*

Sumber: <http://appinventor.mit.edu>

Langkah – langkah yang harus dilakukan dalam pembuatan aplikasi android dengan koneksi Wi-Fi adalah sebagai berikut:

- a. Unduh dan instal aplikasi MIT AI2 *Companion* di Ponsel
- b. Sambungkan komputer dan perangkat dengan jaringan *WiFi* yang sama
- c. Buka proyek *App Inventor* dan hubungkan ke perangkat

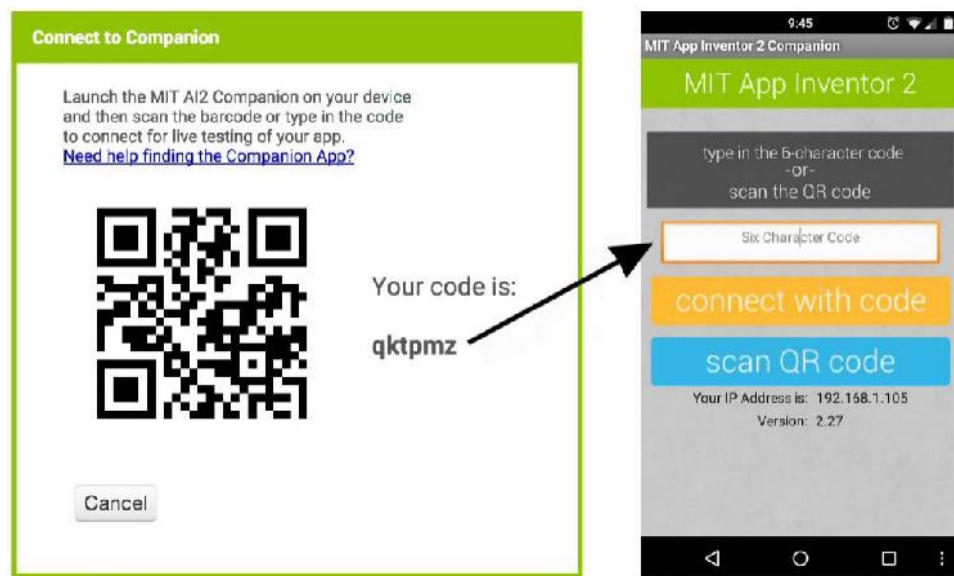
Setelah ketiga langkah diatas selesai, selanjutnya buka Aplikasi *Inventor* kemudian pilih "*Connect*" dan "*AI Companion*" dari menu teratas di browser AI2:



**Gambar 2.7** *Connect AI Companion*

Sumber: <http://appinventor.mit.edu>

Setelah proses diatas selesai selanjutnya secara otomatis akan muncul tampilan kode QR pada halaman MIT Inventor lalu scan kode QR dengan aplikasi yang telah di unduh sebelumnya dari *google playstore*. Setelah semua proses selesai secara otomatis aplikasi yang telah dibuat akan terhubung ke perangkat android.



**Gambar 2.8** Scan QR code

Sumber: <http://appinventor.mit.edu>

## 2. Membuat aplikasi dengan *Emulator* pada PC (*Personal Computer*)

Langkah – langkah yang harus dilakukan dalam pembuatan aplikasi dengan emulator pada PC yaitu dengan menginstal *software App Inventor 2* pada PC. Aplikasi ini dapat berjalan disemua sistem operasi windows dan menggunakan *search engine google chrome* dan *mozilla firefox*. Jalankan *software* yang sudah di *install* sebelumnya, lalu buka *search engine* dan masuk ke halaman *project App Inventor* dan buka *emulator*

### 3. Membuat aplikasi dengan Android dan USB

Membuat aplikasi dengan Android dan USB dapat dilakukan dengan menghubungkan perangkat dengan kabel USB lalu menjalankan koneksi pada halaman kerja App Inventor agar mentransfer data ke perangkat.

## BAB III

### METODE PENELITIAN

#### 3.1 Tahapan Penelitian

Tahapan penelitian pada skripsi ini digambarkan dengan siklus RAD (*Rapid Application Development*).



**Gambar 3.1** RAD (*Rapid Application Development*)

Berikut penjelasan dari siklus RAD (*Rapid Application Development*) :

1. *Requirements Planning* (Perencanaan Syarat – Syarat)

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi dan sistem serta untuk mengidentifikasi syarat - syarat informasi yang ditimbulkan dari tujuan alat dibuat. Orientasi pada fase ini adalah mendapatkan informasi dalam pemanfaatan sistem kendali untuk mengontrol lampu jarak jauh menggunakan perintah suara.

## 2. RAD *design workshop*

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Selama *workshop* design RAD, pengguna merespon *prototype* yang ada dan penganalisis memperbaiki modul-modul yang direncanakan berdasarkan respon pengguna.

## 3. Implementasi

Pada fase Implementasi ini, penerapan penggunaan sistem kendali untuk mengontrol lampu jarak jauh menggunakan perintah suara.

### 3.2 Metode Pengumpulan Data

Pada perancangan dan penyusunan skripsi ini, digunakan beberapa metode untuk mengumpulkan data, diantaranya sebagai berikut:

#### 1. Literatur (Studi pustaka)

Studi pustaka dilakukan untuk mendapat referensi mengenai dasar keseluruhan alat yang digunakan pada sistem, seperti prinsip kerja alat pengontrol lampu jarak jauh menggunakan perintah suara, dan Arduino Uno sebagai Mikrokontoller yang digunakan. Studi pustaka juga digunakan untuk mendapatkan ide tentang pengembangan selanjutnya dalam upaya perbaikan terhadap sistem yang sudah berjalan.

#### 2. Wawancara dan Observasi

Instrumen penelitian dengan teknik wawancara dan observasi, dilakukan dengan cara mengumpulkan serangkaian data terkait dengan permasalahan

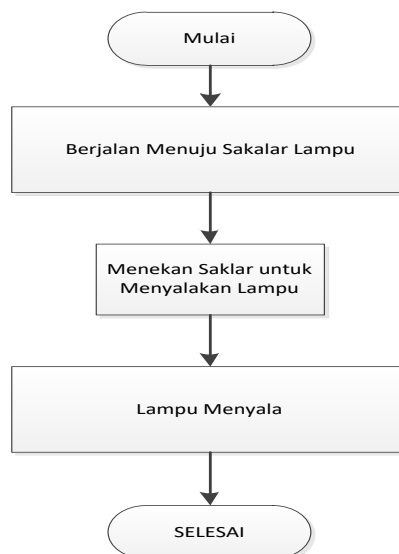
yang sering terjadi pada alat pengontrol lampu jarak jauh menggunakan perintah suara.

### 3.3 Analisis Sistem yang Sedang Berjalan

Analisis sistem merupakan gambaran tentang sistem yang saat ini sedang berjalan. Sistem yang digunakan masih menggunakan cara manual, yaitu menghidupkan lampu dengan menjangkau saklar. Analisis sistem ini bertujuan untuk membuat sistem yang baru agar terkomputerisasi sehingga dapat lebih efektif dan efisien.

#### 3.3.1 *Flow chart* Sistem yang Sedang berjalan

*Flowchart* sistem yang sedang berjalan merupakan diagram alur kerja dari sebuah sistem yang sedang berjalan. Pada penelitian ini, *flowchart* sistem yang berjalan di gambarkan pada gambar 3.2 dibawah ini.



**Gambar 3.2** Flowchart Sistem yang Berjalan

Pada *flowchart* di atas, masih terdapat beberapa kekurangan dengan sistem konvensional saat ini dikarenakan pengguna harus menjangkau saklar lampu jika ingin menghidupkan lampu. Hal ini dirasa kurang efisien karena menghabiskan waktu dan tenaga pengguna. Selain itu, menghidupkan lampu yang dilakukan secara manual terbatas jika dilakukan oleh lansia atau orang dengan keterbatasan menjangkau saklar.

### **3.3.2 Analisis Sistem yang ditawarkan**

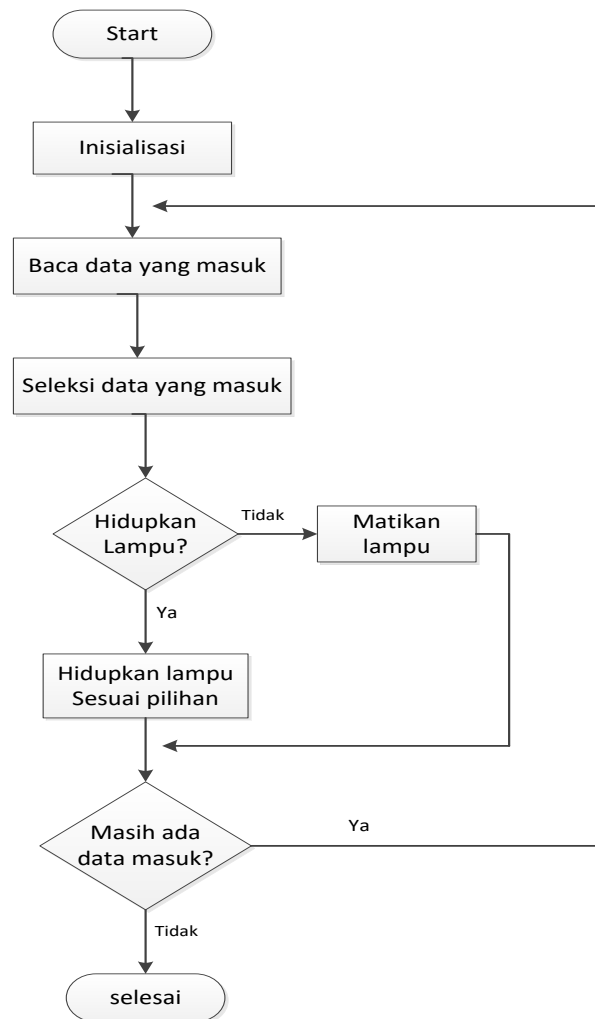
Analisis sistem yang di tawarkan bertujuan untuk mendapatkan solusi baru dalam hal menanggapi setiap masalah dan kekurangan yang masi terjadi pada sistem yang sedang berjalan saat ini. Sistem yang di tawarkan menjawab setiap permasalahan yang ada dengan sistem yang sudah terkomputerisasi dengan sistem kendali untuk mengontrol lampu jarak jauh menggunakan perintah suara. Alat ini menggunakan mendeteksi perintah suara yang diberikan untuk mengontrol lampu secara jarak jauh.

### **3.3.3 *Flowchart* Sistem yang ditawarkan**

*Flowchart* sistem yang ditawarkan menampilkan kegiatan yang akan dilakukan oleh alat pengontrol lampu. Alat yang dirancang mendeteksi perintah suara yang masuk untuk melakukan pengontrolan lampu secara jarak jauh. Pengguna akan memberikan perintah suara melalui aplikasi pada hp android, dimana Android telah terhubung dengan *Arduino Uno* melalui *Bluetooth* sebagai media transmisinya. Suara sebagai input perintah dideteksi oleh *google voice*.



Suara disimpan dan direkam oleh *microphone* pada *handphone*. Suara yang direkam dikirimkan ke server google melalui internet, kemudian google melakukan pembacaan dan memproses suara manusia. Dengan cara melakukan sampling pada suara yang sebelumnya berupa sinyal analog menjadi bit-bit biner (digital). Selanjutnya bit-bit tersebut diproses di dalam database google dan diterjemahkan dalam bentuk text. Hasil pembacaan tersebut dikirimkan kembali ke aplikasi pada Handphone, sehingga aplikasi pada *handphone* dapat menampilkan hasil pembacaan perintah suara



**Gambar 3.3** Flow Chart Sistem yang Ditawarkan

### 3.4 Rancangan Penelitian

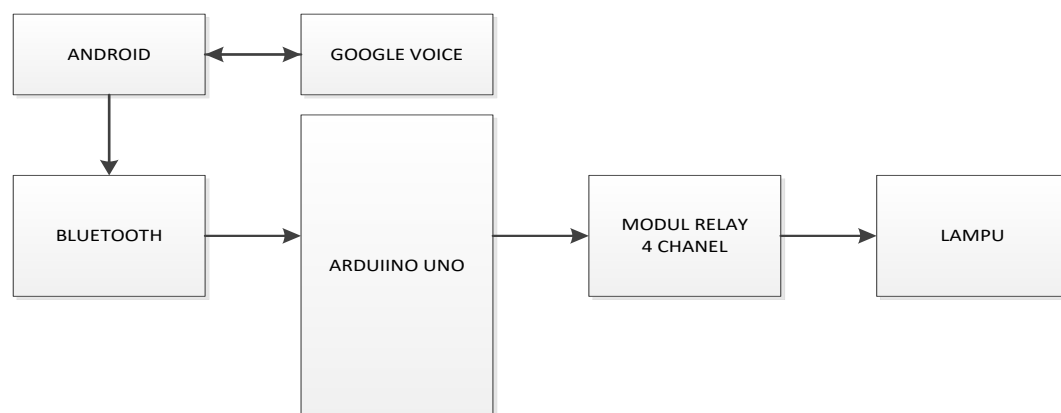
Rancangan penelitian menggambarkan alur dari setiap tahapan yang harus di kerjakan dalam penelitian.

#### 3.4.1 Perancangan Perangkat Keras (*Hardware*)

Agar mempermudah penulis dalam menjelaskan perancangan perangkat keras, maka di gambarkan alur dan cara kerja perangkat keras pada rangkaian diagram blok bawah ini :

##### 3.4.1.1 Diagram Blok Sistem

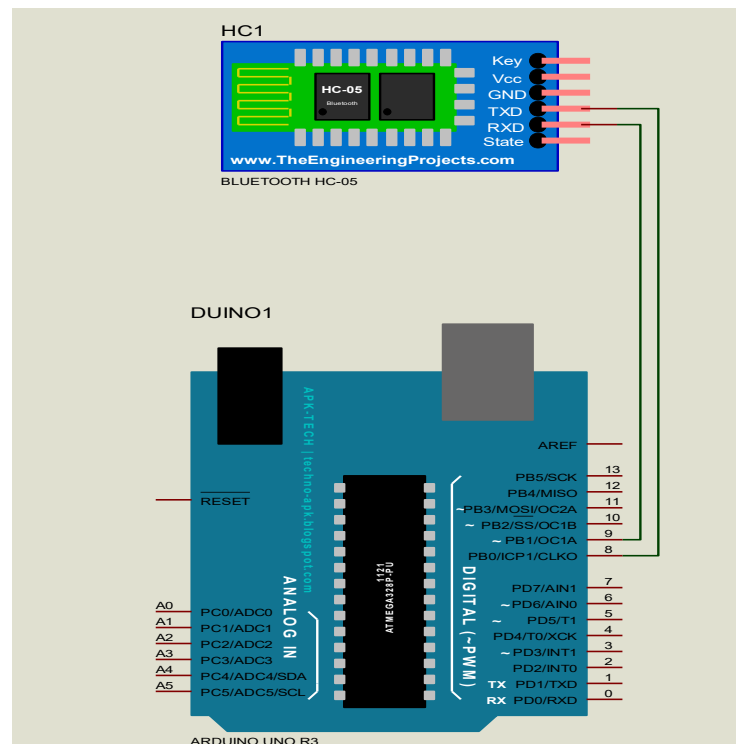
Di dalam merancang dan membuat sistem, terlebih dahulu dilakukan perencanaan blok diagram sehingga skema rangkaian keseluruhan menghasilkan sistem yang baik. Diagram blok merupakan gambaran dasar dari rangkaian sistem yang akan dirancang. Setiap diagram blok mempunyai fungsi masing-masing. Gambar diagram blok dari rangkaian yang akan dirancang seperti pada gambar 3.4. Rangkaian terdiri dari rangkaian modul Bluetooth HC-05, modul Arduino Uno, dan rangkaian driver relay.



**Gambar 3.4** Blok Diagram

Pada gambar 3.4, Smartphone android akan menerima input suara dari user, lalu gelombang suara tadi akan diubah dalam bentuk bilangan *hexadecimal* yang kemudian akan di kirimkan melalui *internet* ke *google voice* untuk di terjemahkan apa arti dari kata yang telah disampaikan oleh pengguna. Selanjutnya, hasil pembacaan data tersebut akan di transmisikan melalui Bluetooth HC-05 yang terkoneksi dengan Arduino uno. Arduino uno akan mengolah data yang dia terima dan akan menjalankan perintah dengan menyalakan atau mematikan relay yang dituju berdasarkan perintah suara yang di terima.

### 3.4.1.2 Rangkaian Bluetooth HC-05

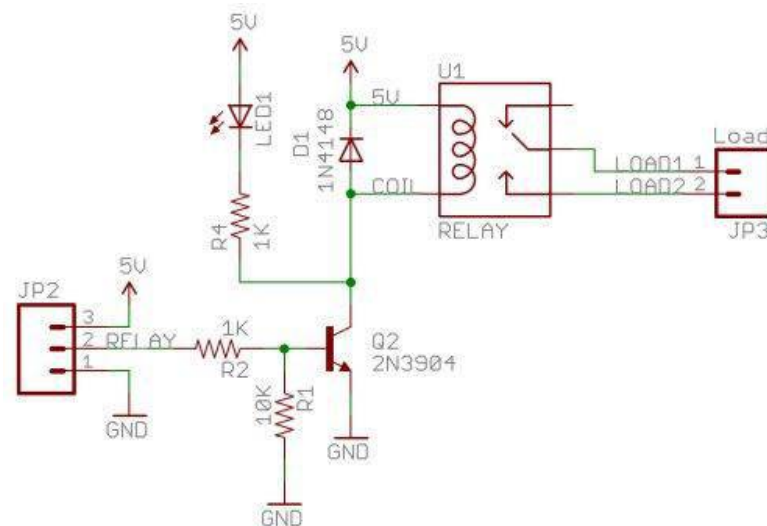


Gambar 3.5 Rangkaian *Bluetooth* HC-05

Pada perancangan ini rangkaian *Bluetooth* HC-05 dapat dilihat sesuai dengan Gambar.3.5, dimana *Bluetooth* bekerja pada tegangan 5 volt yang dihubungkan ke pin 5 volt Arduino uno. Pin GND dihubungkan ke GND Arduino uno. Pin *Receiver* (rx) modul *Bluetooth* HC-05 dihubungkan ke pin PB 1 Arduino uno dan *Tranceiver* (tx) modul *Bluetooth* HC-05 dihubungkan ke pin PB 0 Arduino uno.

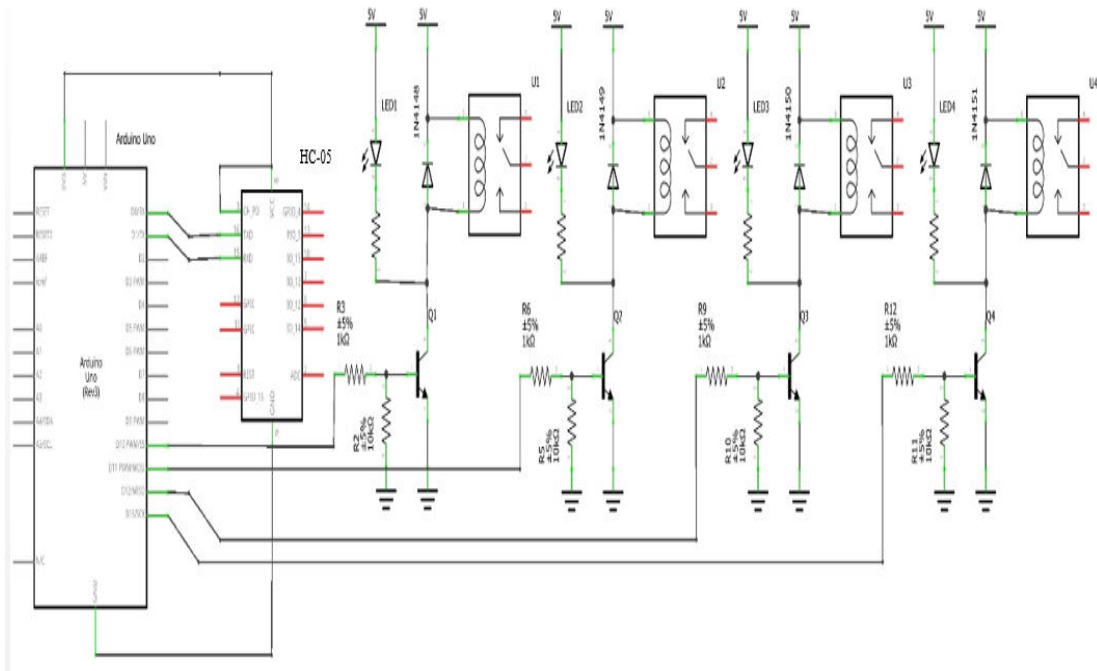
### 3.4.1.3 Rangkaian Modul Relay

Gambar 3.6 menampilkan rangkaian keseluruhan dari sebuah *modul relay*, dimana pada JP2 , port 2 kita sebut sebagai IN (*input*) pada *modul relay* yang akan menerima masukan dari Arduino Uno. Pada bagaian keluaran *relay*, *Load 1* dihubungkan dengan arus PLN sedangkan *Load 2* di hubungkan secara serial pada beban (Lampu) yang kemudian di hubungkan pula dengan *wireline* PLN lainnya.



**Gambar 3.6** Rangkaian Modul Relay

### 3.4.4 Rangkaian Keseluruhan



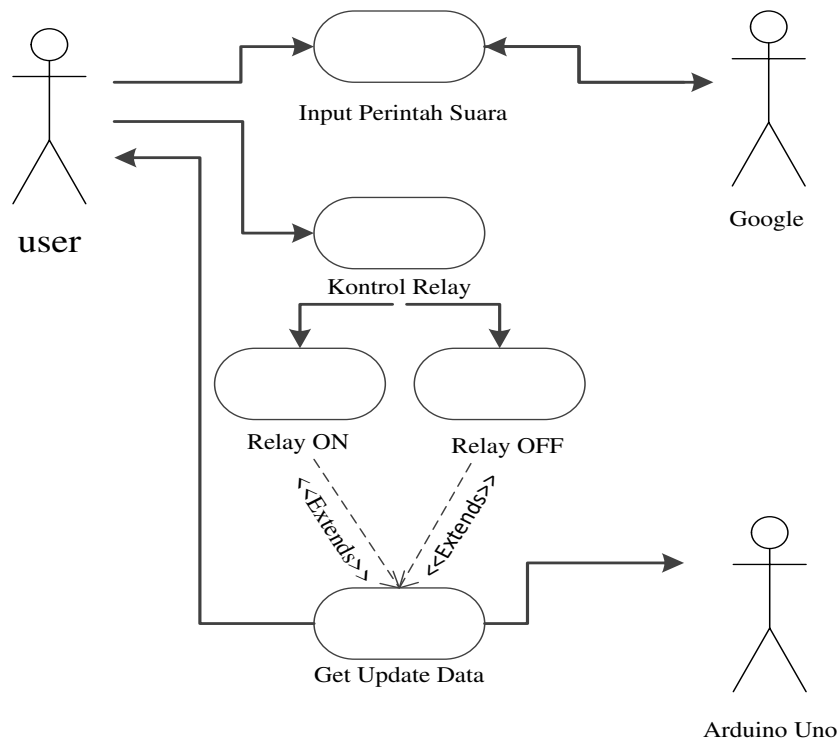
**Gambar 3.7** Rangkaian Keseluruhan

Skema dari keseluruhan rangkaian ditampilkan pada Gambar 3.7. Hubungan antara satu perangkat dengan perangkat yang lain di rancang sesuai dengan program yang akan dijalankan berdasarkan *flowchart* yang telah dibuat.

### 3.4.2 Perancangan Perangkat Lunak (Software)

Pada perancangan ini, perangkat lunak yang dimaksudkan adalah *interface* yang akan berhubungan langsung dengan pengguna, yaitu *smartphone android*. Perancangan perangkat lunak juga dimaksudkan untuk memberikan perintah pada *EPROM* Arduino uno untuk mengeksekusi setiap data yang diterima untuk memberikan *output* pada *relay* apakah berada pada posisi menyala atau mati.

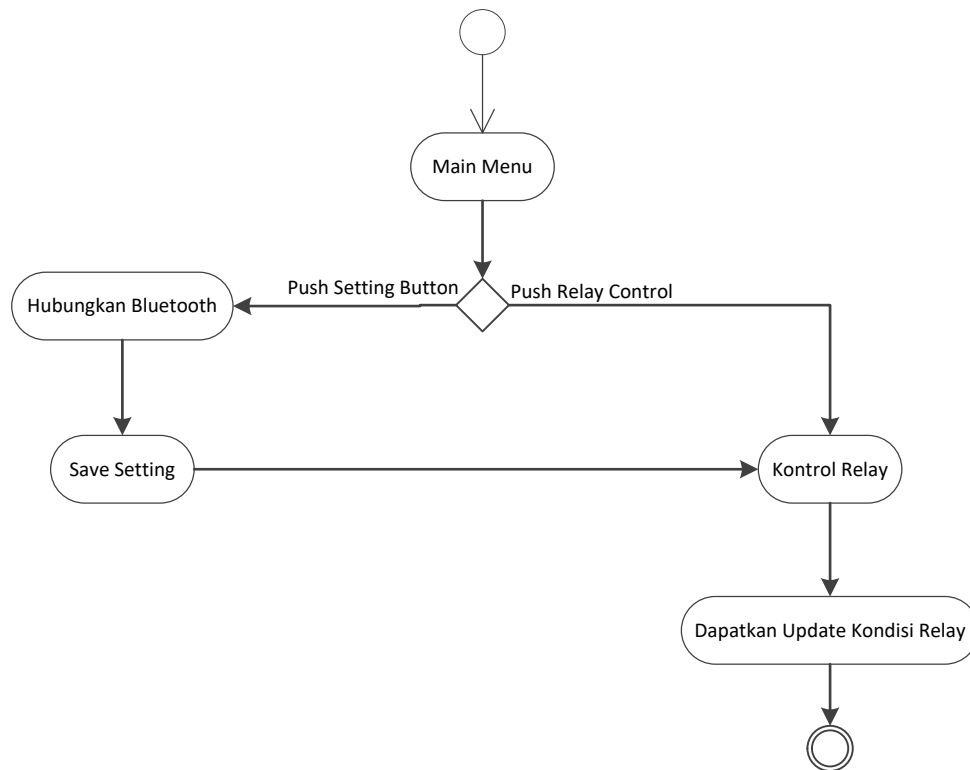
### 3.4.2.1 Use Case Diagram



**Gambar 3.8** Use Case Diagram

Gambar 3.8 menampilkan *diagram use case* dari keseluruhan rancangan pada penelitian ini. Interaksi yang terjadi antara *user* ke sistem adalah dengan memberikan input suara pada aplikasi android yang telah dibuat, dimana aplikasi tersebut sudah terkoneksi dengan database *google voice* yang dapat mendeteksi pita suara manusia dan menerjemahkan apa yang disampaikan oleh *user* yang kemudian akan ditransmisikan oleh *bluetooth* yang terhubung antara *smartphone android* dengan mikrokontroler Arduino Uno. Arduino uno berfungsi sebagai pusat kendali yang akan mengubah posisi *relay*, apakah berada pada posisi menyala atau mati, bergantung pada *input* yang diterima dalam bentuk pesan suara.

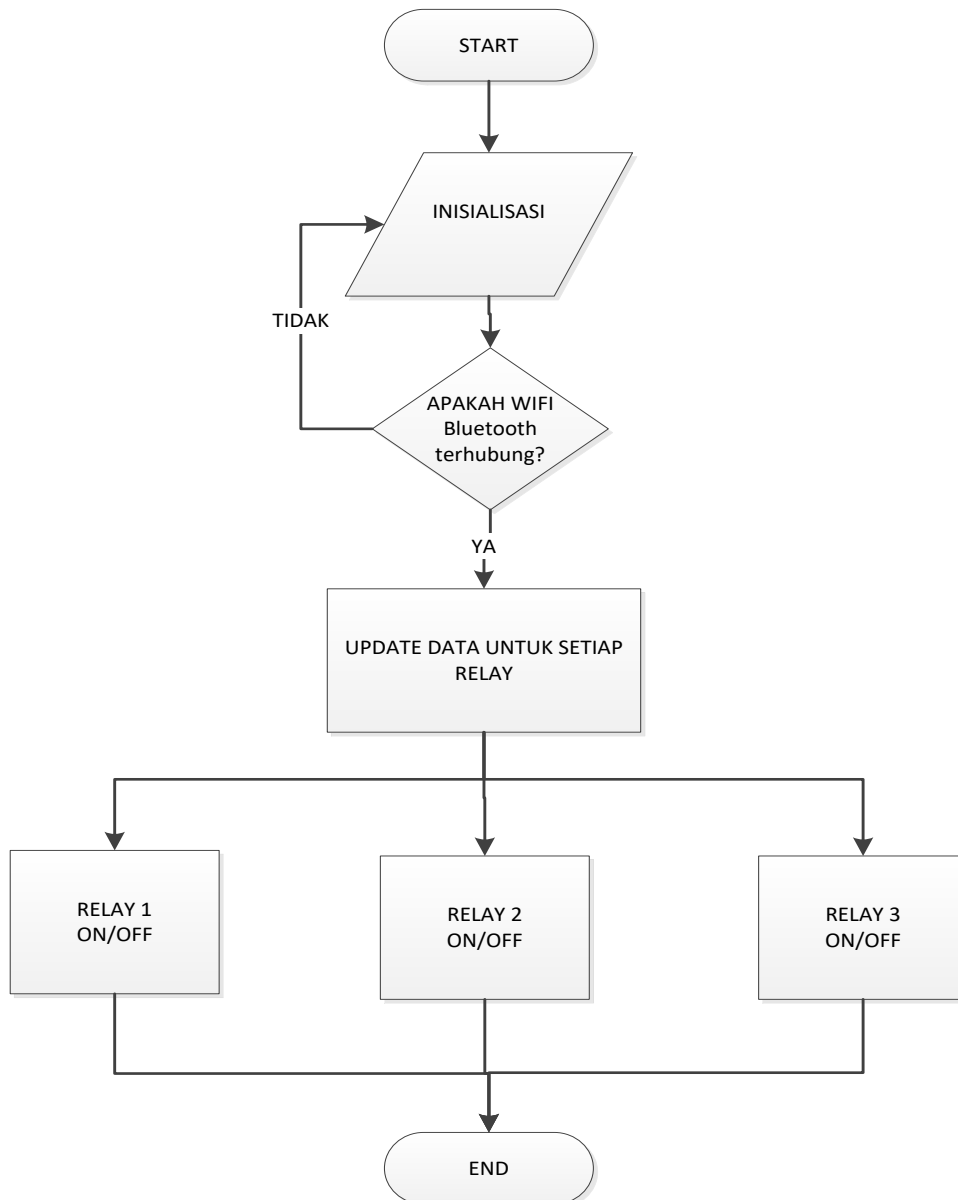
### 3.4.2.2 Activity Diagram



**Gambar 3.9** Activity Diagram

Diagram *Activity* menampilkan alur kerja dari keseluruhan sistem dimana pada saat aplikasi pertama kali dibuka akan muncul tampilan main menu, kemudian user harus mengarahkan pilihan pada tombol *setting* untuk melakukan koneksi antara *Bluetooth* yang ada pada *Arduino Uno* dengan *Bluetooth* yang tertambat di *Smartphone Android*. Setelah proses koneksi berhasil dilakukan, aplikasi yang telah dibuat berada pada posisi *standby* menunggu perintah untuk mengontrol *relay* menyalakan lampu yang berasal dari *input* suara pengguna. Saat lampu berhasil dinyalakan, maka *Arduino* akan mengirimkan data terupdate berkaitan dengan kondisi *relay* mana yang berhasil menyala atau mati dimana data tersebut akan ditampilkan pada aplikasi android.

### 3.4.2.3 Flow Chart Hardware



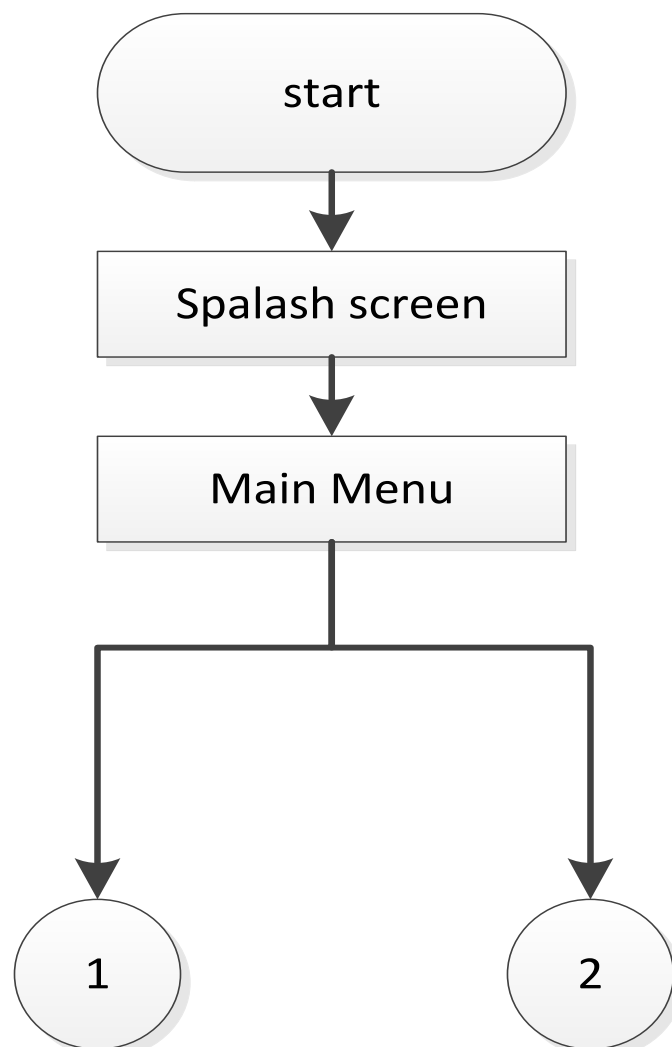
**Gambar 3.10** Flow chart Arduino Uno

Pada skripsi ini, *flow chart* yang akan digunakan dalam membangun sebuah program berbasis Arduino uno terlihat seperti pada gambar 3.10. Pada gambar diatas, setelah *Bluetooth* berhasil melakukan inisialisasi, maka *Bluetooth*

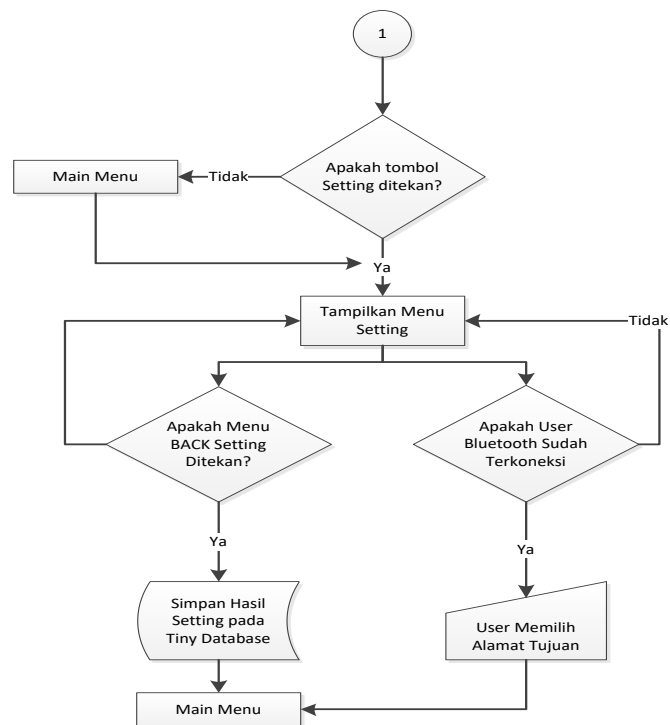


akan menunggu terkoneksi kepada *Smartphone Android* lalu mengirimkan data teraktual berkaitan dengan posisi dari lampu, apakah berada pada posisi menyala atau tidak. Kemudian, semua berada pada kondisi *stanby* hingga ada *input* suara yang diberikan yang nantinya akan diteruskan kepada Arduino Uno untuk diolah menjadi sebuah perintah agar lampu menyala dan sebaliknya.

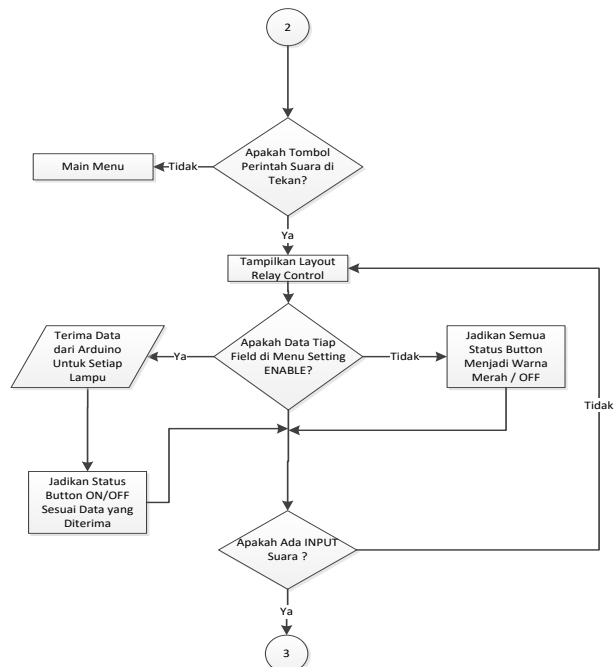
#### 3.4.2.4 Flow Chart Android



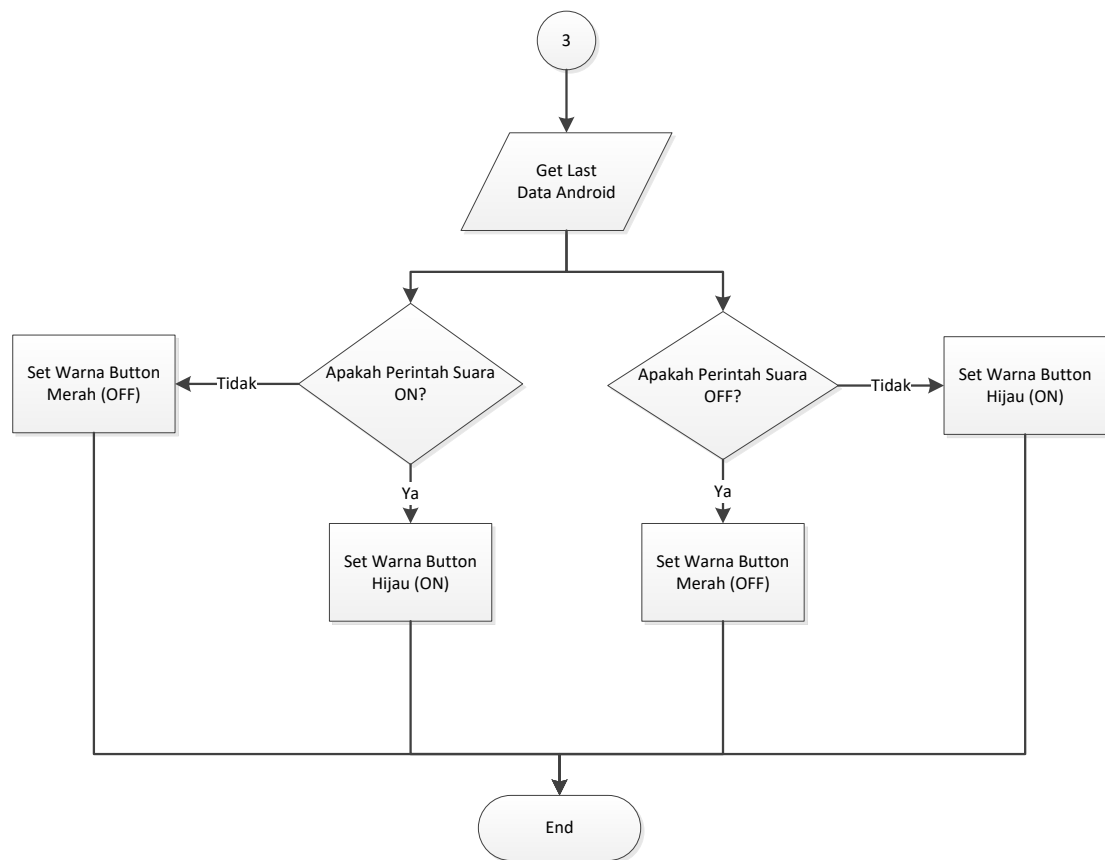
**Gambar 3.11** *Flow chart Android*



Gambar 3.12 Flow chart Android No(1) Lanjutan



Gambar 3.13 Flow chart Android No(2) Lanjutan



**Gambar 3.14** Flow chart Android No(3) Lanjutan

*Flowchart android* menampilkan alur program dari aplikasi yang akan dibuat. Untuk setiap pertama kali dijalankan, maka akan menuju pada main menu. Pada layar main menu tersedia dua bagian yaitu, setting dan kontrol suara. Saat *setting* di tekan maka akan ditampilkan pada halaman untuk melakukan koneksi *Bluetooth* antara hardware mikrokontroler Arduino Uno dengan Smartphone Android. Saat koneksi sudah berhasil dilakukan maka aplikasi akan dikembalikan pada tampilan main menu dan dapat langsung menggunakan halaman kontrol suara.

Menu kontrol suara sudah ditambahkan agar berkomunikasi dengan *google voice* dan dapat dijalankan apabila *Smartphone android* telah terkoneksi dengan

internet. Jika tidak ada koneksi internet maka aplikasi tidak dapat dijalankan. Saat *input* suara diterima oleh *Smartphone android*, secara otomatis signal data suara akan di transmisikan ke *server google voice* untuk diterjemahkan, kemudian hasil penerjemahan tadi akan dikembalikan ke Android yang kemudian akan diolah menjadi perintah untuk menyalakan dan mematikan relay.

### **3.4.3 Lokasi Perancangan**

Perancangan ini dilakukan di laboratorium fakultas Ilmu Komputer Universitas Pembangunan Pancabudi Medan yang beralamat di Jalan Gatot Subroto simpang Sei Sekambing, Kota Medan, Sumatera Utara.

### **3.4.4 Langkah Pengujian Alat**

Pada penelitian ini pengujian dilakukan dengan beberapa tahapan diantaranya adalah sebagai berikut:

1. Pengujian koneksi

Pengujian koneksi dilakukan untuk mengetahui apakah sistem yang dibuat dapat berhubungan dengan baik. Hubungan yang dimaksud adalah proses pengiriman dan penerimaan data melalui port *receiver* dan port *transceiver* yang terdapat pada modul *Bluetooth HC-05* yang terhubung dengan Arduino Uno. Setelah koneksi antara modul *Bluetooth HC-05* dan Arduino telah berjalan dengan baik, pengujian koneksi selanjutnya adalah dengan menguji koneksi antara sistem mikrokontroller dan smartphone Android. Pengujian dapat dilakukan dengan melihat langsung pengaruh yang akan

diterima bila sistem telah terkoneksi secara langsung dengan *smartphone* android.

## 2. Pengujian Alat Pengontrol Lampu

Pengujian pengontrol lampu dapat dilakukan dengan memberikan perintah suara pada *smartphone* android untuk menghidupkan lampu.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Untuk Mementukan Suatu Penelitian Berkeja atau tidak Maka Dibuatlah Sebuah Gambaran Untuk Mengetahui Apakah Sistem itu Bekerja atau tidak Dalam konteks Perancangan yang akan di uraikan yaitu Kebutuhan dan Spekfikasi pada *Hardware* Dan *Sotfware*, Penjelasan *source code* pada Arduino IDE dan pada link [ai2.app.inventor](https://ai2.app.inventor) untuk melakukan koneksi *google voice* ke perangkat Arduino uno. Pada akhir pembahasan ini akan memperlihatkan *hardware* lampu yang menyala.

#### **4.1 Kebutuhan Spesifikasi Minimum *Hardware* dan *Software***

##### **4.1.1 Kebutuhan spesifikasi minimum *Hardware***

Spesifikasi minimum *hardware* yang digunakan alat kendali lampu bisa berfungsi dengan baik yaitu:

1. Arduino Uno
2. Bluetooth HC-05
3. Relay
4. Fitting Lampu Duduk 3 Buah
5. Bola Lampu Pijar 5 Watt

#### 4.1.2 Kebutuhan spesifikasi minimum *Software*

Spesifikasi minimum *Software* yang digunakan untuk memprogram Alat Kendali Lampu adalah:

1. Link [ai2.appinventor.mit.edu](http://ai2.appinventor.mit.edu)
2. Arduino IDE versi 1.8.2

#### 4.2 Pengujian Aplikasi dan Pembahasan

Pada Bab ini *prototype* yang telah di bangun kemudian dilakukan pengujian dan analisis guna untuk mengetahui kinerja sistem kendali lampu. Pengujian berupa pengolahan *hardware* dan *software* yang telah terintegrasi satu sama lain.

##### 4.2.1 Pengujian Arduino

*Arduino Uno* merupakan pengendali utama dari *hardware* yang di buat. Pengujian terhadap *Arduino* ini yaitu untuk mengetahui apakah mikrokontroler ini dapat digunakan dengan baik atau tidak. Cara menguji *hardware* ini yaitu dengan memeriksa setiap *pin input* dan *output* yang terdapat pada *Arduino* yang sebelumnya telah di pasang program pada setiap *pin*. Hasil pengujian menunjukkan bahwa *Arduino* yang sebelumnya telah di program dan disambungkan dengan *hardware* lainnya dapat berjalan. Adapun *Source Code*-nya dapat di lihat di bawah ini :

```
#include <SoftwareSerial.h>
SoftwareSerial BT(2, 3);
String perintah;
```

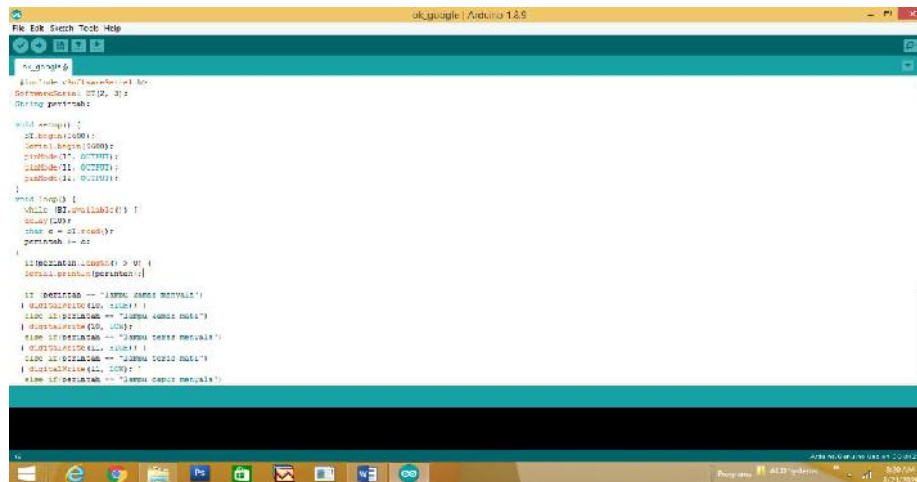
```

void setup() {
  BT.begin(9600);
  Serial.begin(9600);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
}
void loop() {
  while (BT.available()) {
    delay(10);
    char c = BT.read();
    perintah += c;
  }
  if(perintah.length() > 0) {
    Serial.println(perintah);

    if (perintah == "lampu kamar menyala")
    { digitalWrite(10, HIGH); }
    else if(perintah == "lampu kamar mati")
    { digitalWrite(10, LOW); }
    else if(perintah == "lampu teras menyala")
    { digitalWrite(11, HIGH); }
    else if(perintah == "lampu teras mati")
    { digitalWrite(11, LOW); }
    else if(perintah == "lampu dapur menyala")
    { digitalWrite(12, HIGH); }
    else if(perintah == "lampu dapur mati")
    { digitalWrite(12, LOW); }
    else if(perintah == "hidupkan semua")
    { digitalWrite(10, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(12, HIGH); }
    else if(perintah == "matikan semua")
    { digitalWrite(10, LOW);
      digitalWrite(11, LOW);
      digitalWrite(12, LOW);}
    perintah="";} }

```

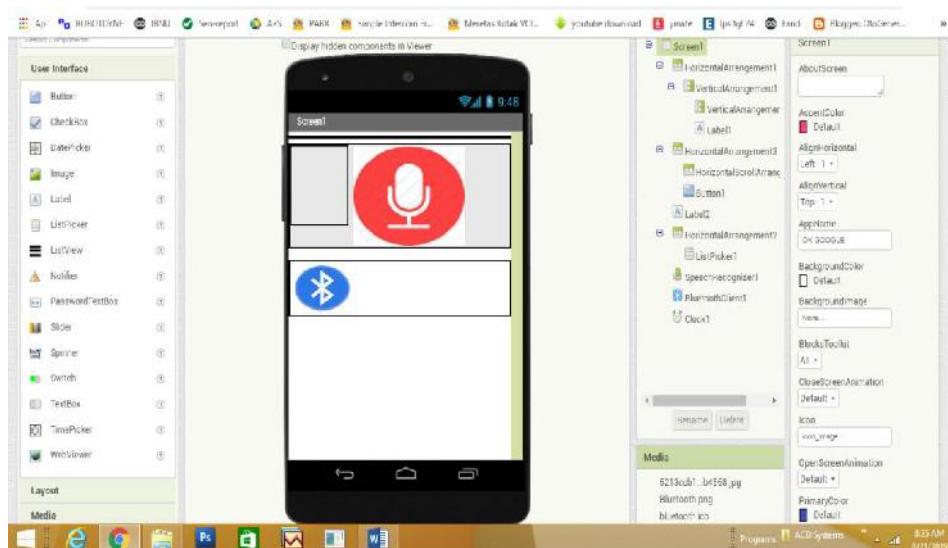




Gambar 4.1 Tampilan Pemrograman Arduino

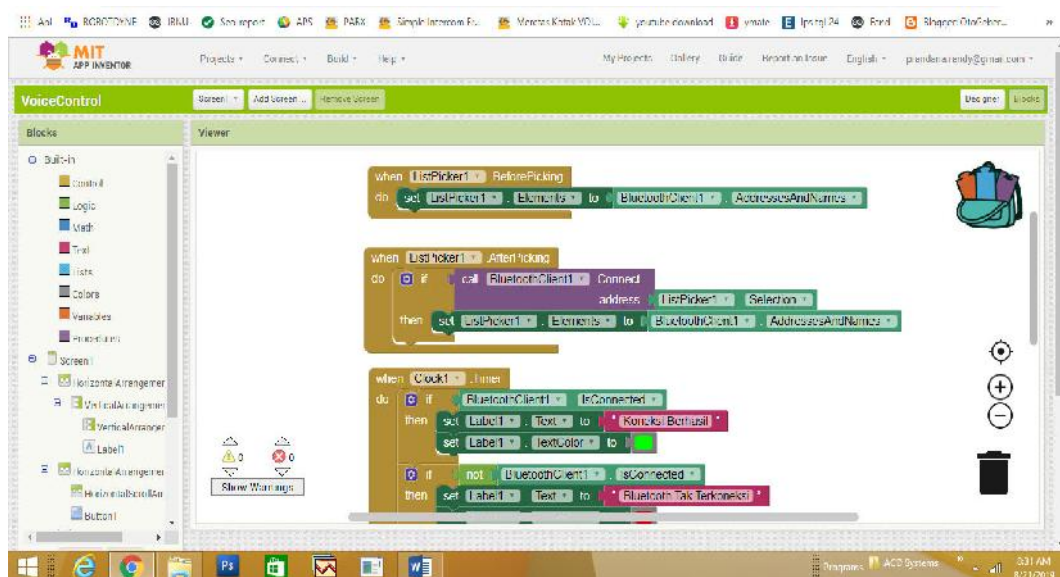
#### 4.2.2 Pengujian Aplikasi Android

Pengujian design android digunakan untuk menguji hubungan yang terjadi antara *google voice* dan perangkat Arduino uno. Dibawah ini merupakan gambaran dari design aplikasi yang telah di rancang



Gambar 4.2 Tampilan Design Aplikasi

Pada *design* aplikasi digunakan beberapa komponen *user interface* yang nantinya akan diisi oleh program pada setiap fungsi yang di inginkan. Pada perancangan aplikasi android digunakan *button* sebagai tombol untuk melakukan koneksi *Bluetooth* antara *smartphone android* dan perangkat *Arduino uno*. *Button* yang lain digunakan untuk memanggil fungsi *google voice* yang secara otomatis akan terhubung ke *database google* untuk menerjemahkan setiap bahasan maupun kalimat yang diucapkan. Fungsi *clock* digunakan sebagai pewaktu *realtime* saat komunikasi data sedang berlangsung. Fungsi *Bluetooth* digunakan untuk menampilkan *address Bluetooth* yang terdapat pada lingkungan sekitar *smartphone android*. Untuk penjelasan lebih lanjut mengenai fungsi dari setiap *user interface* yang dibuat ditampilkan pada gambar dibawah ini:



**Gambar 4.3** Tampilan Pemrograman Aplikasi Android

### 4.2.3 Pengujian Relay

Pengujian ini dilakukan apakah *relay* bekerja dengan normal. Jika *relay* di berikan *output 1* dari *arduino* maka *relay* dalam keadaan *No (Normaly Open)*

begitu juga sebaliknya jika *relay* diberikan *output* 0 maka *relay* dalam keadaan Nc (*Normaly Close*).

**Tabel 4.1.** Tabel pengujian *relay* pada alat kendali lampu

<i>Relay</i>	<i>Data</i>	<i>Solenoid Valve</i>	<i>Status</i>
	0	Tertutup	Lampu Mati
	1	Terbuka	Lampu Hidup

#### 4.2.4 Pengujian Bluetooth

Pada Alat kontrol pengendali lampu di gunakan *Bluetooth* sebagai alat koneksi antara android dan arduino. Pengujian *Bluetooth* di lakukan untuk mengetahui seberapa jauh jarak maksimal *Bluetooth* mampu terhubung dengan alat kontrol. Berikut Pengujian *Bluetooth* berdasarkan jarak (m) :

**Tabel 4.2.** Tabel pengujian bluetooth pada alat kendali lampu

Percobaan	Jarak (m)	Hasil	Delay
1	5	Hidup	1 Detik
2	10	Hidup	1 Detik
3	20	Hidup	1 Detik
4	25	Hidup	1 Detik
5	30	Tidak Hidup	-

Pengujian yang telah di lakukan pada *Bluetooth* HC-05, untuk mengetahui kemampuan *Bluetooth* terkoneksi ke perangkat pengendali lampu dapat diketahui kemampuan *Bluetooth* mampu terkoneksi pada jarak terjauh 25 meter. Pengujian dilakukan pada lapangan terbuka dan daya listrik didapat dari listrik rumah tangga.

#### 4.2.5 Pengujian Seluruh Sistem

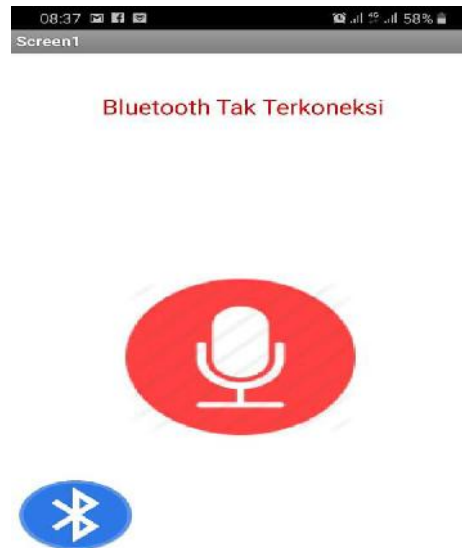
Pengujian ini di lakukan untuk mengetahui apakah seluruh sistem hardware dan *software* yang telah di buat dan di rangkai pada alat kendali lampu dapat tersinkronisasi dan bekerja dengan baik. Berikut tabel hasil pengujian seluruh sistem alat kendali lampu .

**Tabel 4.3.** Tabel pengujian seluruh sistem pada alat kendali lampu

<b>Nama</b>	<b>Posisi Awal</b>	<b>Mengirim Data</b>	<b>Delay Jaringan</b>	<b>Posisi akhir</b>	<b>Hasil</b>
Lampu 1	Mati	1 Detik	2 Detik	Hidup	Berhasil
Lampu 2	Mati	1 Detik	2 Detik	Hidup	Berhasil
Lampu 3	Mati	1 Detik	2 Detik	Hidup	Berhasil
Lampu 4	Mati	1 Detik	2 Detik	Hidup	Berhasil

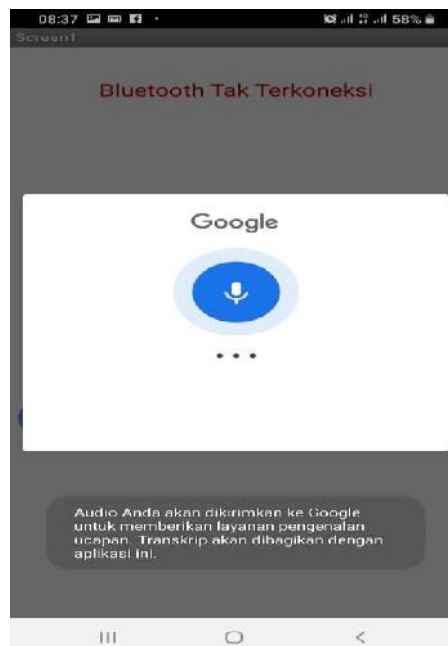
#### 4.2.6 Tampilan Interface

Tampilan antar muka ini berisi satu halaman utama yaitu kontrol alat kendali lampu. *Interface* yang di gunakan untuk mengontrol lampu di hubungkan melalui koneksi *Bluetooth*. Pada tampilan *interface* pengguna dapat menghidupkan dan mematikan lampu dengan memberikan perintah suara sesuai dengan perintah yang sebelumnya sudah disiapkan untuk diterima oleh perangkat Arduino uno.



**Gambar 4.4** Tampilan interface

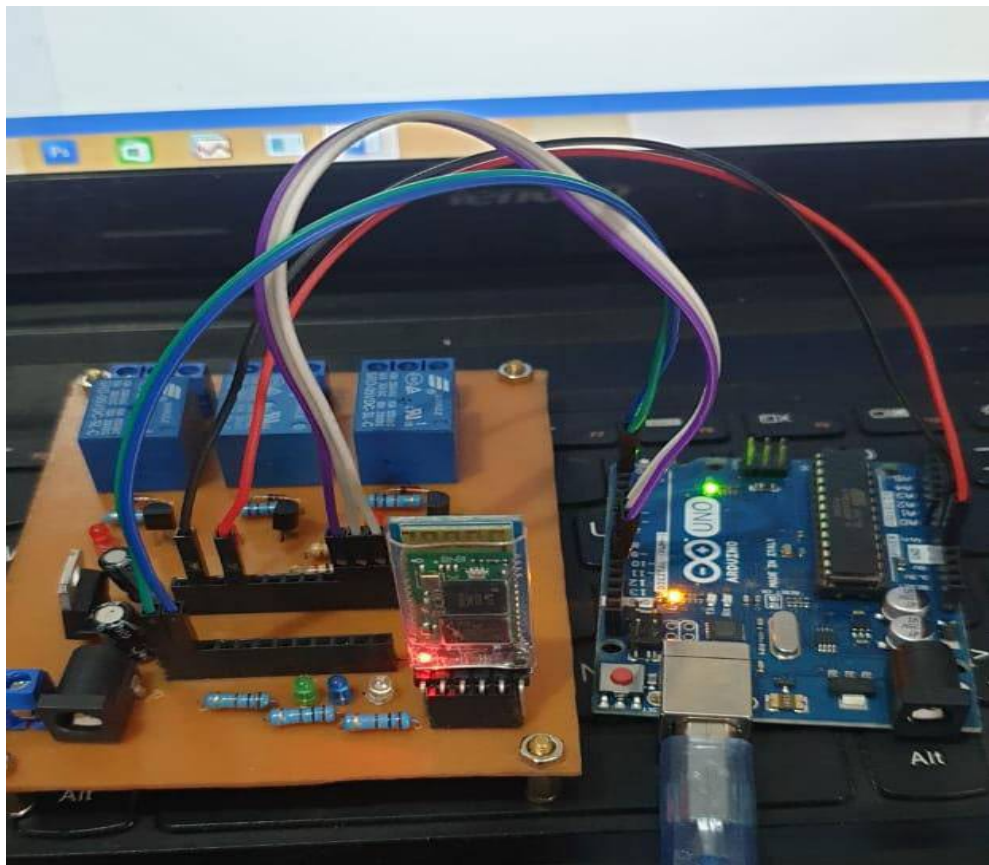
Gambar 4.5 dibawah ini merupakan tampilan saat aplikasi menunggu pesan suara yang akan disampaikan untuk diterjemahkan pada arsip *google voice*.



**Gambar 4.5** Tampilan Koneksi Goole Voice

#### 4.2.7 Pengujian Perancangan ALat Kendali Lampu

Pengujian perancangan alat kendali lampu terdiri dari beberapa komponen, diantaranya adalah perangkat Bluetooth, 3 buah relay dan Arduino uno. Pada perancangan di bawah ini, modul Bluetooth dapat berkoneksi dengan baik dengan smartphone android.



**Gambar 4.6** Tampilan Perangkat Keras

User dapat menghidupkan dan mematikan lampu dengan menggunakan perintah suara, percobaan dilakukan untuk menguji hasil apakah sesuai dengan yang penulis rancang. Hasil dari pengujian dapat dilihat pada gambar berikut:



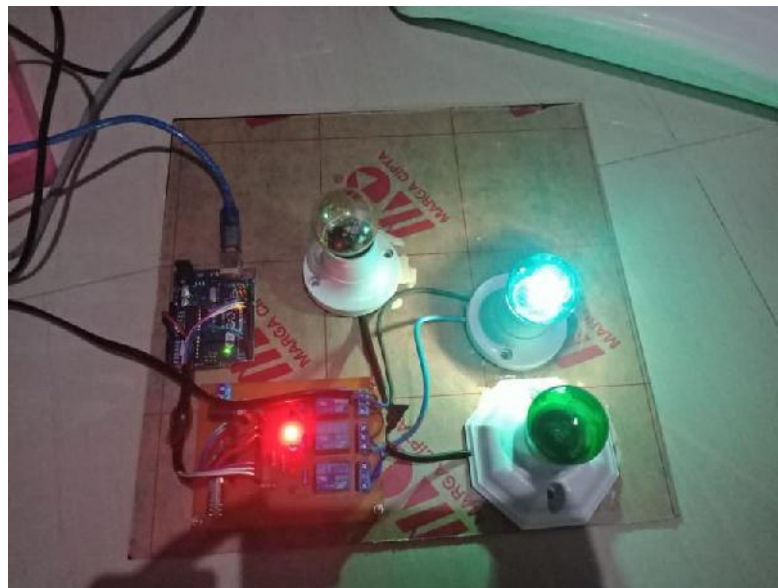
**Gambar 4.7** Tampilan Aplikasi Saat Perintah Menyalakan Lampu Dapur



**Gambar 4.8** Lampu Dapur Menyala

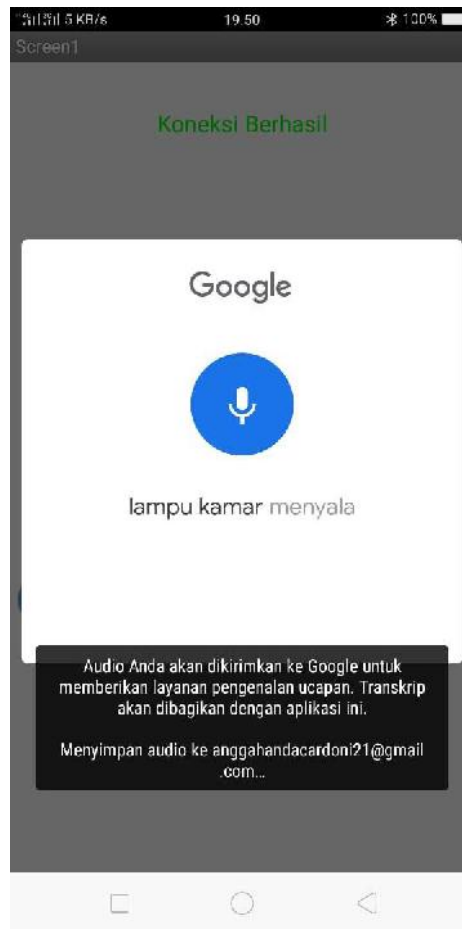


**Gambar 4.9** Tampilan Aplikasi Saat Perintah Menyalakan Lampu Teras

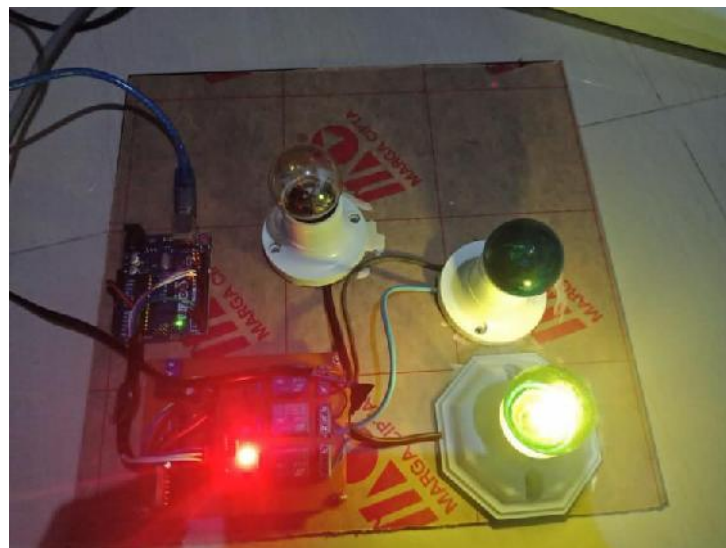


**Gambar 4.10** Lampu Teras Menyala

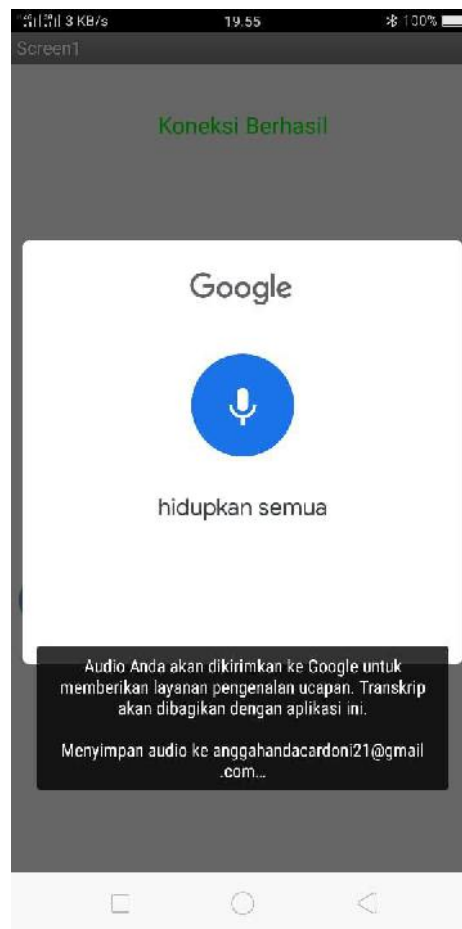




**Gambar 4.11** Tampilan Aplikasi Saat Perintah Menyalakan Lampu Kamar



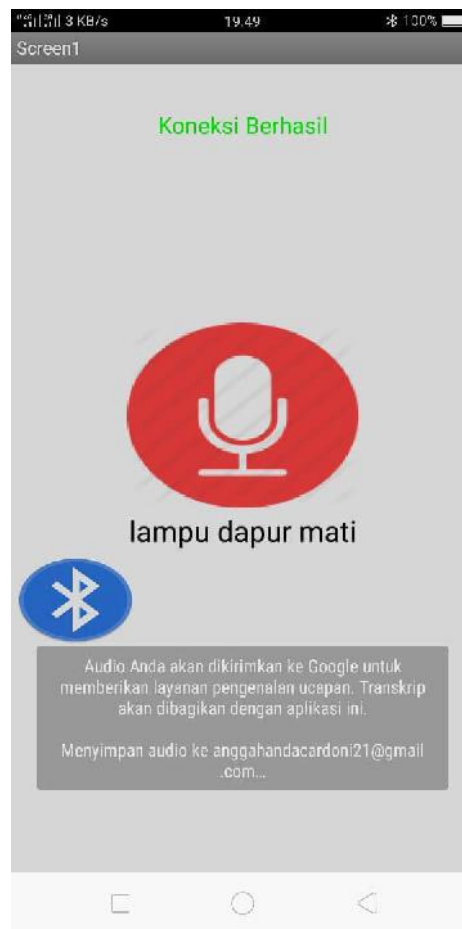
**Gambar 4.12** Lampu Kamar Menyala



**Gambar 4.13** Tampilan Aplikasi Saat Perintah Menyalakan Semua Lampu



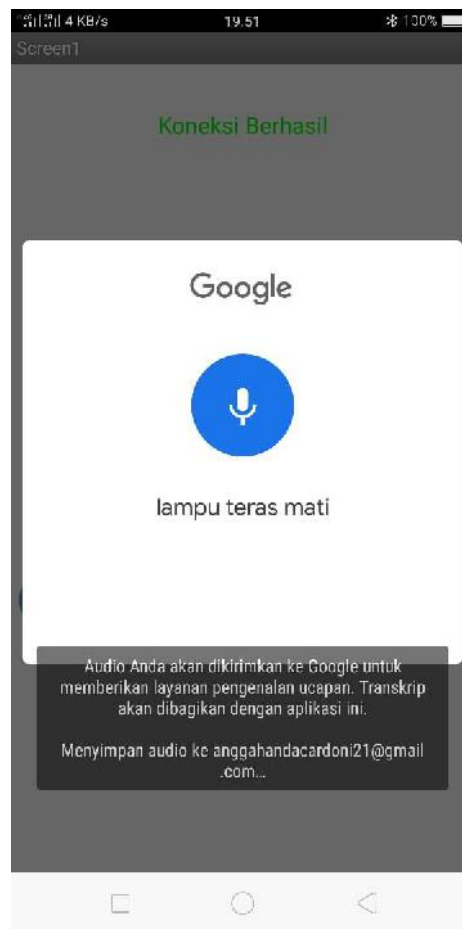
**Gambar 4.14** Seluruh Lampu Menyala



**Gambar 4.15** Tampilan Aplikasi Saat Perintah Mematikan Lampu Dapur



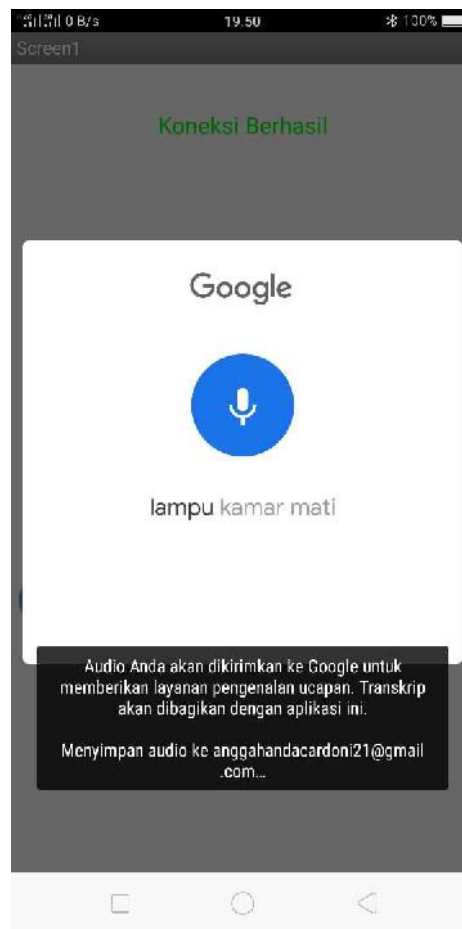
**Gambar 4.16** Lampu Dapur Mati



**Gambar 4.17** Tampilan Aplikasi Saat Perintah Mematikan Lampu Teras



**Gambar 4.18** Lampu Teras Mati



**Gambar 4.19** Tampilan Aplikasi Saat Perintah Mematikan Lampu Kamar



**Gambar 4.20** Lampu Kamar Mati



**Gambar 4.21** Tampilan Aplikasi Saat Perintah Mematikan Semua Lampu



**Gambar 4.22** Seluruh Lampu Mati

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil penelitian yang dilakukan oleh penulis, maka penulis dapat menarik beberapa kesimpulan.

1. Aplikasi yang di buat dapat menghidupkan dan mematikan lampu rumah dengan menggunakan android.
2. Aplikasi yang di buat dengan mengembangkan aplikasi *OKE Google*
3. Media yang di gunakan untuk mentransferkan perintah dari aplikasi ke *prototype* pengendali lampu adalah *Bluetooth*.
4. Batas maksimum Jangkauan *Bluetooth* untukk terkoneksi pada pengendali lampu adalah 25 meter.

#### **5.2 Saran**

Adapun Saran Penulis untuk Sistem Untuk Pengedali Lampu Menggunakan Bluetooth adalah hanya bisa terkoneksi dengan jarak 25 meter seharusnya bisa di integrasikan dengan internet jadi dapat digunakan pada jarak jauh.

## DAFTAR PUSTAKA

- Anonim, E. H. Rachmawanto and C. A. Sari, "Keamanan File Menggunakan Teknik Kriptografi Shift Cipher," *Jurnal Techno. Com*, vol. 14, no. 2, pp. 329-338, 2017.
- Badawi, A. (2018). Evaluasi Pengaruh Modifikasi Three Pass Protocol Terhadap Transmisi Kunci Enkripsi.
- Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). *Jurnal Media Informatika Budidarma*, 2(2).
- Bishop, Rosdiana, "*Sekuritas Sistem Dengan Kriptografi*," in *Prosiding Sendi\_U 2013*, Semarang, 2013.
- Dhany, H. W., Izhari, F., Fahmi, H., Tulus, M., & Sutarman, M. (2017, October). Encryption and decryption using password based encryption, MD5, and DES. In *International Conference on Public Policy, Social Computing and Development 2017 (ICOPOSDev 2017)* (pp. 278-283). Atlantis Press.
- Dhany, H. W., Izhari, F., Fahmi, H., Tulus, M., & Sutarman, M. (2017, October). Encryption and decryption using password based encryption, MD5, and DES. In *International Conference on Public Policy, Social Computing and Development 2017 (ICOPOSDev 2017)* (pp. 278-283). Atlantis Press.
- Fresly, Faizal Zuli1, Ari Irawan, "*Implementasi Kriptografi Dengan Algoritma Blowfish dan Riverst Shamir Adleman (RSA) Untuk Proteksi File*," *Jurnal Format* Volume 6 nomor 2 Tahun 2016.
- Fuad, R. N., & Winata, H. N. (2017). Aplikasi Keamanan File Audio Wav (Waveform) Dengan Terapan Algoritma Rsa. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 1(2), 113-119.



- Gede Angga Pradipta " *Penerepan Kombinasi metode Enkripsi Vigenere Cipher Dan Trasposisi Pada Aplikasi Client Server Chatting*, " *Jurnal Sistem Dan Informatika* Vol. 10, Nomor 2, 2016.
- Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 1(1).
- Hendrawan, J. (2018). Rancang Bangun Aplikasi Mobile Learning Tuntunan Shalat. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 44-59.
- Iqbal, M., Siahaan, A. P. U., Purba, N. E., & Purwanto, D. (2017). Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning. *Int. J. Sci. Res. Sci. Technol*, 3(6), 504-509.
- Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. *Jurnal Teknik dan Informatika*, 5(2), 13-19.
- Nandar Pabokory, Indah Fitri Astuti, Awang Harsa Kridalaksana, " *Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard*," *Jurnal Informatika Mulawarman* Vol. 10. Nomor 1, 2015.
- Putri, N. A. (2018). Sistem Pakar untuk Mengidentifikasi Kepribadian Siswa Menggunakan Metode Certainty Factor dalam Mendukung Pendekatan Guru. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 78-90.
- Rahim, R. (2018, October). A Novelty Once Methode Power System Policies Based On SCS (Solar Cell System). In *International Conference of ASEAN Prespective and Policy (ICAP)* (Vol. 1, No. 1, pp. 195-198).
- Ramadhan, A., & Mohd. Awal Hakimi. (2006). *Pemrograman Web Database dengan PHP dan MySQL*. Synergy Media.
- Ramadhan, M., & Nugroho, N. B. (2009). Desain web dengan php. *Jurnal Saintikom*, 6(1).
- Renddy, Teady Matius, Surya Mulyana, Fresly, " *Steganografi Dengan Deret Untuk Mengacak Pola Penempatan Pada Rgb*," *Jurnal Teknologi Informasi*, 2015.

- Rhee, C. A. Sari, E. H. Rachmawanto, Y. P. Astuti and L. Umaroh, "Optimasi Penyandian File Kriptografi Shift Cipher," in Prosiding Sendi\_U 2013, Semarang, 2013.
- Sarif, M. I. (2017). Penemuan Aturan yang Berkaitan dengan Pola dalam Deret Berkala (Time Series).
- Sitorus, Z. (2018). Kebutuhan Web Service untuk Sinkronisasi Data Antar Sistem Informasi dalam Universitas. *Jurnal Teknik dan Informatika*, 5(2), 87-90.
- Sumartono, I., Siahaan, A. P. U., & Mayasari, N. (2016). An overview of the RC4 algorithm. *IOSR J. Comput. Eng*, 18(6), 67-73.
- Suriski Sitinjak, Yuli Fauziah, Juwairiah, " *Aplikasi Kriptografi File Menggunakan Algoritma Blowfish*," *Jurnal Informatika Mulawarman* Vol. 10. Nomor 1, 2015.
- Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 100-109.