

## BAB IV

### HASIL DAN PENGUJIAN

#### 4.1 Hasil Perancangan

Pada tahapan setelah selesai perancangan yang penulis lakukan adalah menganalisa hasil alat yang telah dibuat. Dalam pembuatan alat ini terbagi menjadi dua bagian pembuatan, yaitu pembuatan *hardware* yang berupa perangkat mikrokontroler *Arduino Uno* sebagai pemroses data dan pengatur beban dan pembuatan *software smartphone* yang digunakan sebagai perangkat *input* pengendali. Adapun hasil pembuatan alat dan *hardware* dan *software* aplikasi *android* adalah sebagai berikut :

a. Hasil Pembuatan *Hardware*

Dalam hasil pembuatan *hardware* ini penulis membuat desain *box* yang digunakan sebagai perangkat pemroses dan *driver* beban. Dengan penggunaan *box* ini akan meningkatkan dari segi keamanan kemudian mempermudah dalam pengoperasiannya dan dapat terlihat lebih rapi pada saat digunakan. Adapun desain dan hasil *hardware* yang sudah dibuat oleh penulis adalah sebagai berikut

:



Gambar 4.1 *Box* tampak depan dan atas



Gambar 4.2 *Box* tampak belakang dan atas

Pada gambar *box* diatas terdapat tulisan-tulisan keterangan pada tombol ataupun indikator untuk memudahkan dalam pengoperasian alat ini. Pada bagian belakang *box* terdapat port untuk *input* tegangan AC 220V dan pada bagian depan terdapat dua soket *output* ke beban 220 V AC.



Gambar 4.3 *Box* tampak samping kiri dan atas

Pada gambar diatas yaitu pada gambar *box* sebelah kiri terdapat barcode , barcode tersebut merupakan barcode aplikasi *android* yang sudah penulis buat. Apabila barcode tersebut di scan maka akan langsung diarahkan ke internet untuk langsung bisa mendownload aplikasi pada perangkat *android*.



Gambar 4.4 *Box* tampak samping kanan dan atas

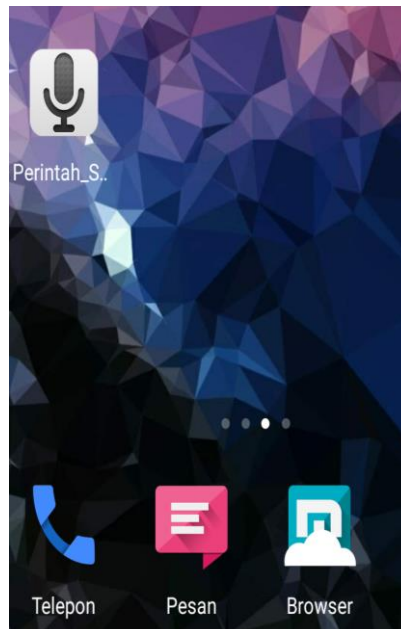
Pada gambar *box* diatas pada sebelah kanan *box* terdapat port USB yang berfungsi sebagai penghubung *Arduino* dengan PC dalam pemrograman atau juga dapat sebagai sumber tegangan luar. Selain terdapat port USB pada sisi kanan *box* juga terdapat PORT *input* DC 9-12V DC sebagai catu daya *eksternal* yang dapat digunakan untuk mengaktifkan alat ini. Pada alat ini terdapat juga *switch* yang berada

di sebelah atas *box* yang memiliki fungsi untuk mengaktifkan pemrograman apabila akan dilakukan pengisian program secara langsung melalui port USB.

b. Hasil pembuatan Aplikasi *android*

Dalam pembuatan aplikasi *android* ini penulis menggunakan software App Inventor pada komputer secara *online*. Secara umum fungsi aplikasi ini adalah sebagai perangkat masukan perintah suara dengan mengaktifkan fungsi *speech recognition* kemudian data yang sudah didapatkan dikirimkan ke *Arduino* melalui koneksi *bluetooth* untuk selanjutnya diproses untuk pengaturan beban.

Adapun tampilan aplikasi *android* pada *smartphone* yang penulis buat adalah sebagai berikut :



Gambar 4.5 Tampilan Icon Aplikasi Perintah Suara

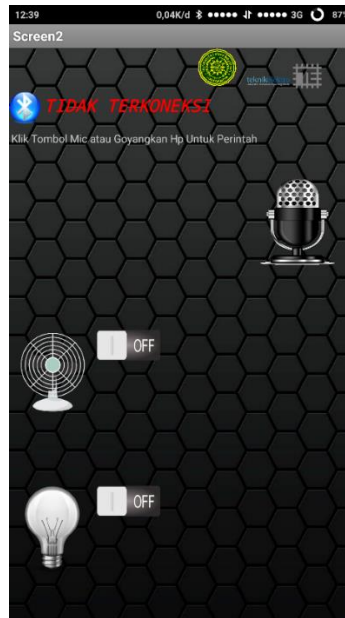


Gambar 4.6 Tampilan awal aplikasi perintah suara

Pada gambar diatas merupakan gambar tampilan awal aplikasi perintah suara pada *smartphone android*. Pada tampilan awal ini belum masuk pada bagian pengontrolan , pada *screen* ini hanya sebagai tampilan awal aplikasi. Pada gambar diatas terdapat dua tombol di bagian bawah layar.

Sebelah kiri merupakan tombol tentang pembuat, yaitu merupakan tombol yang apabila di klik akan memberikan informasi tentang pembuat aplikasi tersebut melalui fungsi *text to speech*. Sedangkan pada tombol sebelah kanan merupakan tombol yang apabila di klik akan masuk pada *screen* berikutnya, yaitu *screen* pada tampilan pengontrolan . Gambar dan tampilan-tampilan pada aplikasi ini dibuat dengan menambahkan gambar ataupun fungsi-fungsi pada *software App inventor*.

Apabila tombol masuk pengontrolan di tekan, maka *screen* pertama akan berpindah menuju *screen* selanjutnya yaitu *screen* pengontrolan. Adapun gambar *screen* pengontrolan pada aplikasi ini adalah sebagai berikut :



Gambar 4.7 Tampilan pengontrolan aplikasi perintah suara

Pada gambar diatas merupakan tampilan layar pada aplikasi perintah suara, tampilan ini merupakan tampilan pengontrolan yang akan digunakan untuk mengontrol beban dengan perintah suara. Pada bagian atas terdapat tombol koneksi *Bluetooth*, apabila *bluetooth* belum terkoneksi maka muncul teks ” *TIDAK TERKONEKSI* “ berwarna merah dan apabila terkoneksi akan muncul tulisan “ *TERKONEKSI* “ berwarna hijau.

Tombol *bluetooth* juga digunakan untuk pairing dengan *bluetooth* pada perangkat pengontrol . kemudian terdapat tombol yang berupa gambar *mic*, tombol ini berfungsi untuk memanggil *speech recognition* sebagai perangkat *input*. Selain

diaktifkan dengan cara di klik fungsi *speech recognition* juga dapat diaktifkan dengan menggoyangkan hp karena penulis mengaktifkan fungsi sensor *accelerometer* pada aplikasi ini.

Pada tampilan dibawah layar terdapat gambar kipas dan lampu dan gambar saklar merupakan indikator yang dapat langsung dilihat pada saat aplikasi berjalan. Indikator tersebut akan aktif pada saat kipas ataupun lampu diperintahkan aktif dan indikator tersebut akan berubah menjadi ON dan berwarna hijau. Dalam pembuatan program *android* ini penulis membuat 9 macam perintah yang dapat diproses oleh aplikasi tersebut untuk mengendalikan beban , keseluruhan perintah dapat dilihat pada tabel berikut :

Tabel 4.1 Daftar Perintah Suara

| NO | PERINTAH                  | KETERANGAN                   |
|----|---------------------------|------------------------------|
| 1  | “ <i>Hidupkan Semua</i> “ | Menghidupkan Lampu dan Kipas |
| 2  | “ <i>Matikan Semua</i> “  | Mematikan Lampu dan Kipas    |
| 3  | “ <i>Lampu Terang</i> “   | Mengaktifkan Lampu Terang    |
| 4  | “ <i>Lampu sedang</i> “   | Mengaktifkan Lampu Sedang    |
| 5  | “ <i>Lampu Redup</i> “    | Mengaktifkan Lampu Redup     |
| 6  | “ <i>Kipas Cepat</i> “    | Mengaktifkan Kipas Cepat     |
| 7  | “ <i>Kipas Lambat</i> “   | Mengaktifkan Kipas Lambat    |
| 8  | “ <i>Matikan Lampu</i> “  | Mematikan Lampu              |
| 9  | “ <i>Matikan Kipas</i> “  | Mematikan Kipas              |

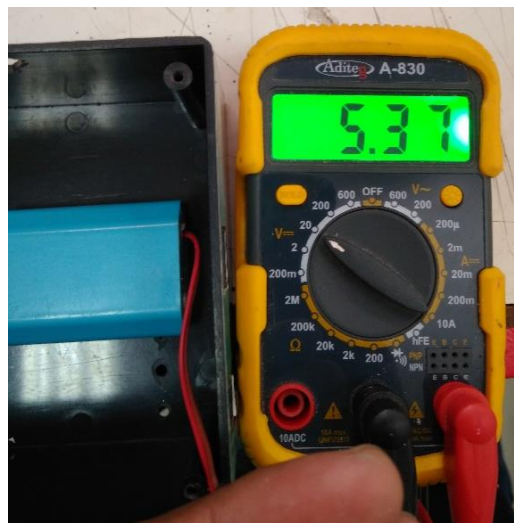
## 4.2 Pengujian Fungsional Alat

Pada tahapan pengujian alat ini dilakukan untuk mengetahui kinerja alat per bagian secara fungsional. Terdapat bagian-bagian dari alat yang perlu diuji secara fungsional sebelum dilakukan pengujian keseluruhan secara uji kinerja. Fungsi dari pengujian ini adalah untuk mengetahui kondisi secara fungsional pada komponen alat. Adapun bagian-bagian yang perlu dilakukan pengujian secara fungsional adalah sebagai berikut :

### a. Pengujian Fungsional *hardware*

#### 1. Adaptor 220 V AC – 5 V DC

Adaptor ini digunakan sebagai pensuplay tegangan 5 V DC ke *Arduino*. *input* adaptor ini langsung tersambung dengan tegangan 220 V AC dan *outputnya* langsung terhubung dengan catu daya pada *Arduino*. Adapun hasil pengujian fungsional dengan mengukur tegangan *output* adaptor menggunakan multimeter digital adalah sebagai berikut :



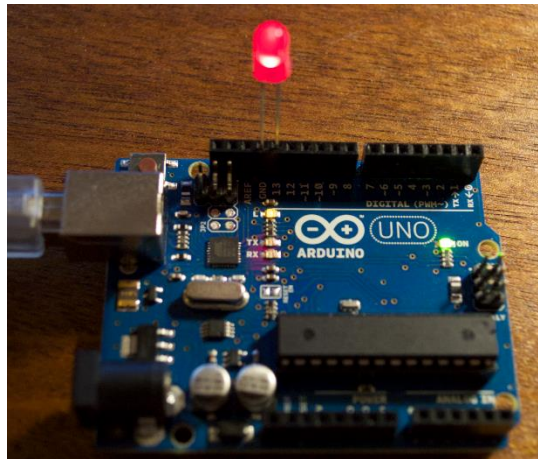
Gambar 4.8 Pengukuran Adaptor



Pada pengukuran adaptor menggunakan multimeter digital terbaca nilai tegangan *output*nya adalah 5,3 V. Dengan demikian *output* tegangan dari adaptor tersebut secara fungsional dapat dikatakan baik karena dapat memberikan *output* tegangan sebesar 5 V sesuai dengan spesifikasinya.

## 2. Arduino Uno R3

Pada pengujian *arduino* ini dilakukan untuk mengetahui apakah kondisi *arduino* masih baik ataupun tidak. Pengujian dilakukan dengan memberikan perintah program *blink* LED pada pin 13. Adapun hasil pengujian dapat dilihat pada gambar berikut :



Gambar 4.9 Pengujian *Arduino*

Pada hasil pengujian *arduino* ini didapatkan hasil bahwa *arduino* masih dalam kondisi baik dibuktikan dengan diberikan perintah menghidupkan led pada pin 13 dan led dapat menyala. Apabila kondisi *arduino* sudah tidak baik maka akan terjadi *error* ataupun tidak dapat dimasukan program.

### 3. *Module Bluetooth HC-05*

Pengujian modul *bluetooth* HC-05 dilakukan untuk mengetahui apakah modul ini dapat bekerja dengan baik. Modul ini pada saat diberikan tegangan maka led indikator akan berkedip cepat dan pada saat sudah terkoneksi led akan berkedip lambat.

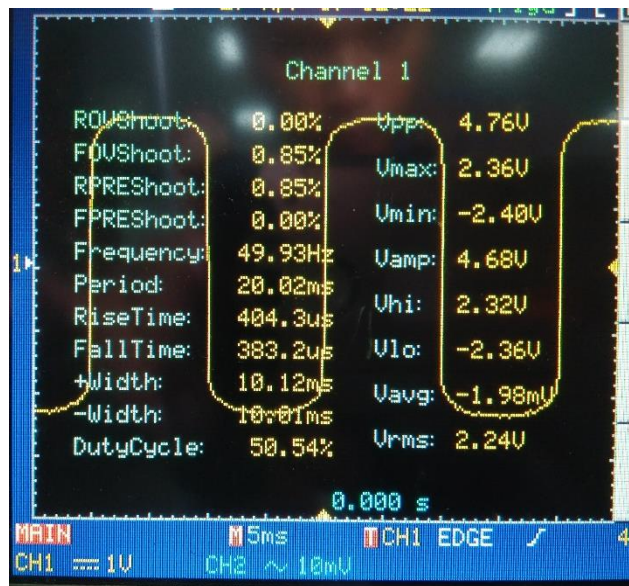


Gambar 4.10 Pengujian HC-05

Dari hasil pengujian modul *bluetooth* HC-05 ini dapat diambil kesimpulan bahwa modul masih dapat bekerja dengan baik dilihat dari indikator led pada saat tidak terkoneksi ataupun saat terkoneksi. Modul *bluetooth* HC-05 ini dapat berfungsi sebagai *master* dan *slave*, namun pada penggunaan di penelitian ini modul HC-05 hanya digunakan sebagai *slave* karena hanya digunakan untuk menerima data yang dikirim dari perangkat *smartphone*.

#### 4. Rangkaian *Zero crossing*

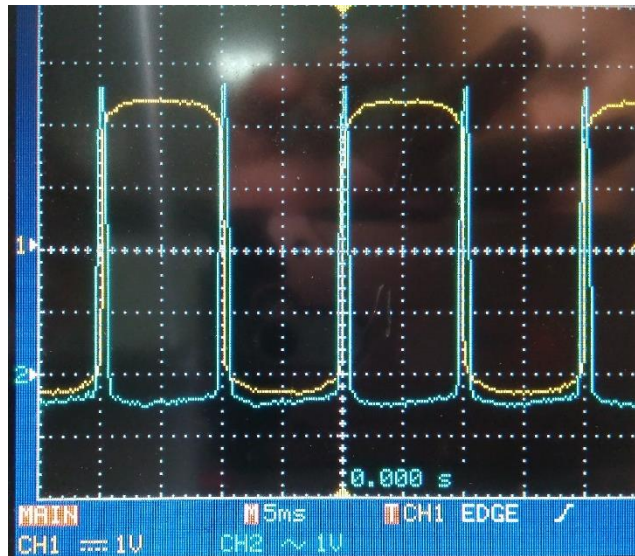
Pengujian rangkaian *Zero crossing* dilakukan untuk mengetahui apakah komponen pada rangkaian *zero crossing* ini dapat bekerja dengan baik ataupun tidak. Rangkaian *zero crossing* pada alat ini hanya digunakan untuk pengaturan kecerahan pada lampu led. Pengujian pada rangkaian *zero crossing* ini dilakukan menggunakan osiloskop digital untuk mengetahui secara lebih detail baik nilai tegangan ataupun sinyal yang dihasilkan. Adapun hasil dari pengujian rangkaian ini adalah sebagai berikut :



Gambar 4.11 Sinyal *input Zero crossing*

Pada gambar diatas merupakan sinyal *input* yang akan masuk pada rangkaian *zero crossing*. Sinyal ini merupakan sinyal dari sumber tegangan AC 220V yang sudah diturunkan tegangannya menggunakan resistor sebesar 30 Kilo ohm.

Sinyal tersebut nantinya akan masuk melalui IC optocoupler untuk mendapatkan *output* sinyal *zero crossing*nya. Gambar dari sinyal *output* rangkaian *zero crossing* adalah sebagai berikut :

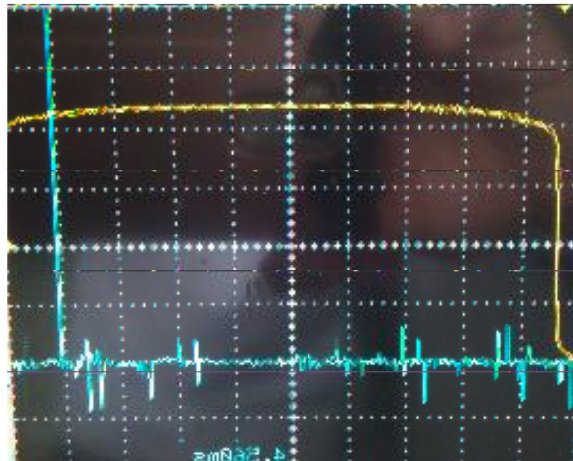


Gambar 4.12 Sinyal *output Zero crossing*

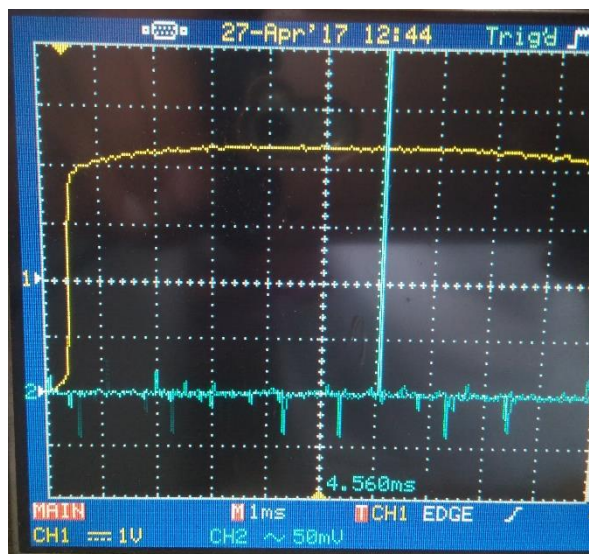
Pada gambar diatas sinyal *output zero crossing* adalah yang berwarna biru. *Output* rangkaian *zero crossing* dapat dikatakan baik karena dilihat dari sinya yang keluar dapat kondisi on pada saat ada persilangan sinyal di titik nol.

Kemudian pengujian sinyal selanjutnya adalah dengan membandingkan sinyal *input* dengan sinyal *output* untuk mendrive TRIAC. Dengan menggunakan osiloskop dengan 2 chanel yang kita aktifkan maka hasil *output* dari kedua sinyal tersebut dapat langsung kita lihat dan kita bandingkan. Pada pengujian ini dilakukan pada saat pemberian perintah untuk lampu pada saat terang kemudian sedang dan selanjutnya lampu led pada saat redup.

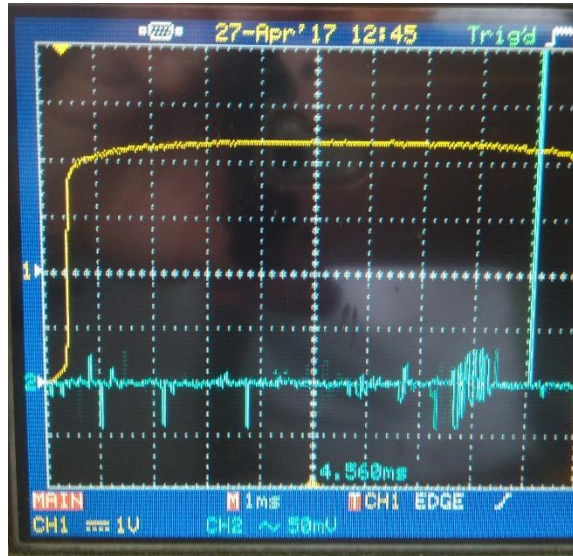
Adapun hasil gambaran sinyal pada osiloskop yang didapatkan adalah sebagai berikut :



Gambar 4.13 sinyal *input* TRIAC saat lampu terang



Gambar 4.14 sinyal *input* TRIAC saat lampu sedang



Gambar 4.15 sinyal *input* TRIAC saat lampu redup

Dari hasil pengamatan sinyal *input driver* TRIAC dapat dikatakan sesuai dengan yang diharapkan. Adapun perhitungannya adalah sebagai berikut :

Frekuensi *input* = 50 Hz

$T = 1/F = 1/50 = 20 \text{ ms}$

$20\text{ms} / 2 = 10 \text{ ms}$  ( pembagian dari setengah gelombang sinus )

Lampu terang = TRIAC aktif pada 1,3 ms

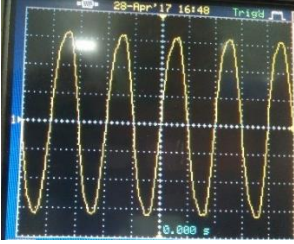
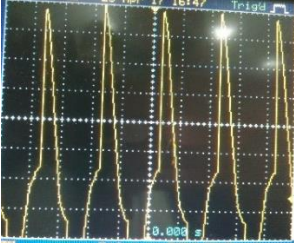
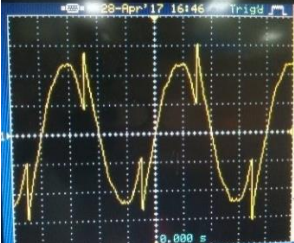
Lampu Sedang = TRIAC aktif pada 7 ms

Lampu Redup = TRIAC aktif pada 8,5 ms

Dengan hasil perhitungan dan hasil pembacaan dari osiloskop yang sesuai maka bisa dikatakan bahwa rangkaian *zero crossing* ini dapat bekerja dengan baik.

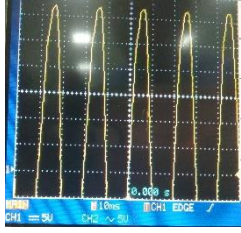
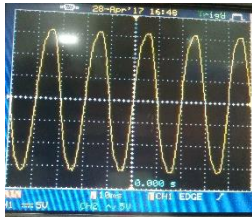
Selanjutnya adalah pengujian *output* pada beban ,pengujian ini dilakukan untuk mengetahui apakah *output* dari rangkaian dapat sesuai dengan yang diharapkan. Untuk pengujian yang pertama adalah pengujian pada *output* beban lampu LED 220V AC dapat dilihat pada tabel berikut :

Tabel 4.2 Pengukuran beban lampu

| BEBAN LAMPU LED 220 VAC 8 W |            |  |
|-----------------------------|------------|--|
| Beban                       | Multimeter | Osiloskop  |
| <i>Lampu Terang</i>         | 216 V      |   |
| <i>Lampu Sedang</i>         | 80 V       |  |
| <i>Lampu Redup</i>          | 30 V       |  |

Pengujian beban selanjutnya adalah pada beban kipas 220 V AC. Adapun hasil dari pengujian beban kipas dapat dilihat pada tabel dibawah ini :

Tabel 4.3 Pengukuran beban kipas

| BEBAN KIPAS 220 VAC 35 W |            |  |
|--------------------------|------------|--|
| Beban                    | Multimeter | Osiloskop  |
| <i>Kipas Cepat</i>       | 217 V      |    |
| <i>Kipas Lambat</i>      | 185 V      |  |

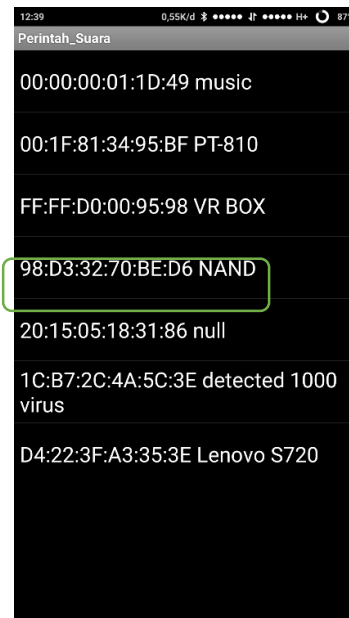
b. Pengujian Fungsional *Software* Aplikasi Perintah Suara *Android*

Pengujian aplikasi *android* dilakukan untuk mengetahui apakah aplikasi yang telah dibuat dapat bekerja dengan baik dan tidak terjadi kendala. Pengujian yang pertama adalah apakah aplikasi ini dapat melakukan pengambilan pencarian nama *bluetooth* yang akan dikoneksikan.

*Bluetooth* HC-05 sudah dirubah namanya oleh penulis melalui AT- Command yaitu namanya menjadi “ NAND ”. Di dalam aplikasi ini dalam programnya bisa memilih nama *Bluetooth* yang akan dikoneksikan.



Adapun gambar tampilan dari layar pada saat pemilihan nama *bluetooth* yang akan dikoneksikan dapat dilihat pada gambar berikut :



Gambar 4.16 Pemilihan nama *Bluetooth*




Gambar 4.17 *Bluetooth* Terkoneksi


Sesuai pada gambar bahwa *bluetooth* dengan nama “NAND” dapat ditemukan, jadi bisa diambil kesimpulan bahwa aplikasi dapat bekerja untuk mencari *bluetooth* dan dapat terkoneksi dengan baik.

Selanjutnya adalah pengujian kompatibilitas aplikasi perintah suara dengan berbagai macam jenis hp dengan sistem operasi *android*. Adapun hasil dari pengujian tersebut dapat dilihat pada gambar dibawah ini :

Tabel 4.4 Spesifikasi Samsung Grand Prime


| Samsung Grand Prime  | Spesifikasi   |
|--|---|
|  | <ul style="list-style-type: none"> <li>- Android Os, V5.1 (Lollipop)</li> <li>- RAM 1GB</li> <li>- Chipset Snapdragon 410</li> <li>- CPU Quad Core</li> </ul> |

Tabel 4.5 Spesifikasi Lenovo S720

| Lenovo S720   | Spesifikasi   |
|---|---|
|  | <ul style="list-style-type: none"> <li>-Android 4.0.4 (Ice Cream Sandwich)</li> <li>-Chipset Mediatek MT6577</li> <li>-CPU Dual-core 1.0 GHz</li> <li>-GPU PowerVR SGX531u</li> </ul> |

Pada pengujian aplikasi di *smartphone* ini dilakukan pengujian pada *smartphone* dengan sistem operasi *android* dari *4 Ice Cream Sandwich* sampai versi yang terbaru. Aplikasi ini dapat berjalan atau dapat di *install* pada sistem operasi *android* versi 4 dan pada versi-versi selanjutnya.

Tabel 4.6 Spesifikasi HP ASUS Zenfone 4s

| ASUS Zenfone 4s  | Spesifikasi   |
|--|---|
|  | <ul style="list-style-type: none"> <li>- O.S. Android OS, v4.4.2 (KitKat)</li> <li>-CPU Intel Atom Z2520 Dual-core 1.2 GHz, GPU PowerVR SGX544MP2</li> <li>-GPU Adreno 330</li> </ul> |

Berdasarkan hasil pengujian aplikasi yang di *install* di beberapa jenis *smartphone android* secara keseluruhan aplikasi dapat berjalan dengan baik, namun ada beberapa jenis *smartphone* yang perlu dilakukan pengaturan dalam fungsi *google voice* dan *text to speech* untuk dapat memberikan *input* masukan pada aplikasi tersebut.

Secara umum aplikasi yang dibuat oleh penulis dapat berjalan pada *smartphone android*, namun beberapa tipe *smartphone* mungkin perlu pengaturan tambahan dan penambahan aplikasi penunjang seperti *google voice* ataupun *text to speech*.

### 4.3 Pengujian Kinerja Alat

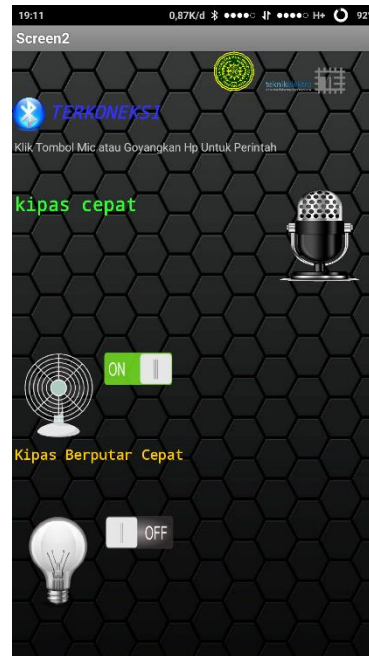
Pada tahap pengujian kinerja alat ini dilakukan untuk mengetahui apakah keseluruhan komponen pada alat ini dan kemampuan alat apakah sesuai dengan yang diharapkan dan juga dapat bekerja dengan baik. Pengujian dilakukan dengan menggunakan alat secara langsung dan kondisi alat sudah terhubung semua. Pengujian dilakukan dengan menggunakan perangkat *software* yang berupa aplikasi *android* maupun dengan *hardware* berupa alat dalam *box* yang sudah dibuat.

Pengujian dilakukan dengan memberikan perintah suara dan melihat pengaruhnya terhadap beban yang dipasangkan sesuai dengan perintah suara yang telah di buat didalam program sebelumnya. Adapun hasil dari pengujian kinerja alat ini adalah sebagai berikut :

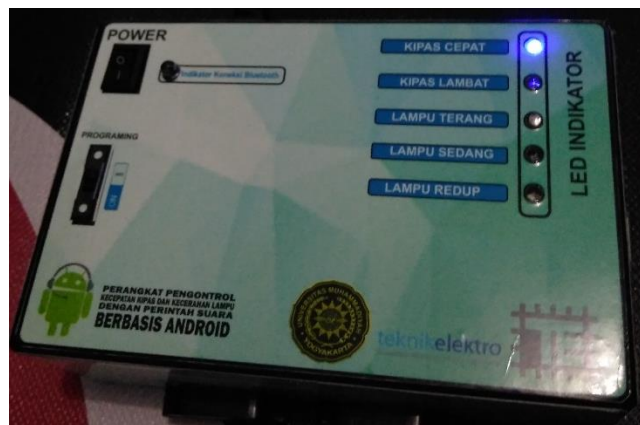
- a. Pengujian kinerja alat pada beban kipas
  1. Pemberian perintah kipas cepat

Pada saat perintah kipas cepat diberikan maka beban yang berupa kipas 220 VAC akan berputar secara maksimal dan indikator led kipas cepat akan menyala. Pada perintah ini *smartphone* akan mengirimkan data melalui *bluetooth* ke *arduino* untuk pengaturan beban kipas untuk mendapatkan tegangan maksimal sehingga putaran kipas dapat cepat.

*Arduino* akan memproses data yang masuk kemudian akan mengatur besaran sinyal PWM yang akan mengatur TRIAC sehingga didapatkan tegangan beban kipas yang maksimal yaitu pada nilai 255 pada PWM. Dengan PWM maksimal maka putaran kipas akan berputar pada putaran paling cepat.

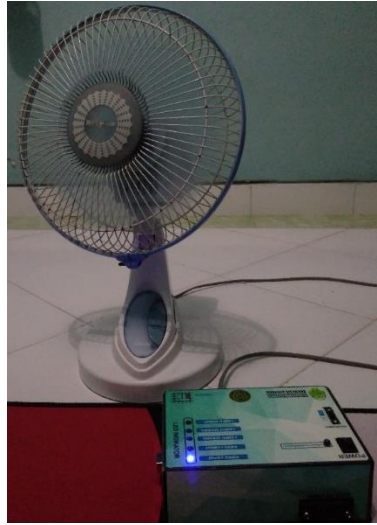


Gambar 4.18 Tampilan aplikasi pada saat kipas cepat



Gambar 4.19 Indikator LED pada saat kipas cepat

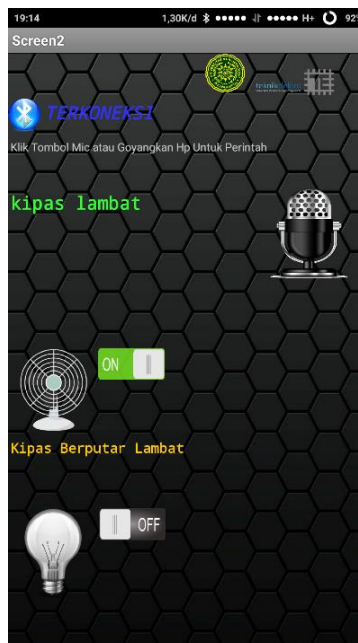
Indikator LED digunakan untuk mengetahui perintah manakah yang sedang dijalankan pada alat ini. Apabila seluruh led indikator mati maka beban juga dalam kondisi *off*. Terdapat 5 jumlah led indikator untuk 3 kondisi beban lampu dan 2 kondisi beban kipas.



Gambar 4.20 kipas berputar cepat

## 2. Pemberian perintah kipas lambat

Pada saat perintah kipas lambat diberikan maka beban yang berupa kipas 220 VAC akan berputar secara tidak maksimal dan indikator led kipas lambat akan menyala.



Gambar 4.21 Tampilan aplikasi pada saat kipas lambat

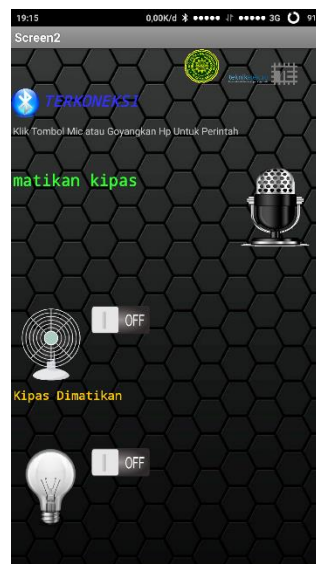
Setelah perintah kipas lambat diberikan , maka akan ada indikator yang ditampilkan pada tampilan aplikasi dan indikator LED pada perangkat beban.



Gambar 4.22 Indikator LED pada saat kipas lambat

### 3. Pemberian perintah matikan kipas

Pada saat perintah ini diberikan maka beban kipas akan berhenti berputar dan akan ada indikator yang tampil pada layar aplikasi ataupun indikator LED pada perangkat beban akan mati.



Gambar 4.23 Tampilan aplikasi pada saat kipas dimatikan



Gambar 4.24 kipas dimatikan

b. Pengujian kinerja alat pada beban lampu

Pada pengujian ini dilakukan menggunakan beban lampu LED 220VAC 8 watt. Pengujian dilakukan untuk dapat mengetahui apakah alat dapat mengontrol beban lampu dengan baik sesuai dengan yang diharapkan. Adapun pengujian yang dilakukan adalah sebagai berikut:

1. Pemberian perintah lampu terang

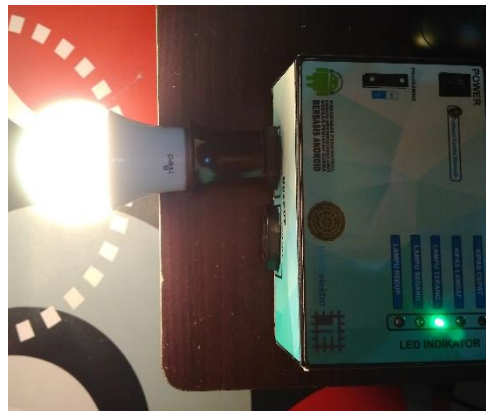
Perintah lampu terang digunakan untuk memberi perintah agar beban pada lampu diberikan tegangan maksimal sehingga cahaya pada lampu didapatkan cahaya yang maksimal.

Dengan perintah tersebut maka *arduino* memberikan sinyal pada *input* TRIAC yang terhubung dengan rangkaian *zero crossing* untuk dapat diaktifkan pada nilai 1.3 ms . Adapun hasil pengujiannya adalah sebagai berikut :





Gambar 4.25 Tampilan aplikasi pada saat lampu terang



Gambar 4.26 Indikator LED pada saat lampu terang

## 2. Pemberian perintah lampu sedang

Pada pemberian perintah lampu sedang ini bertujuan untuk memberikan output tegangan menengah pada beban lampu sehingga cahaya lampu yang dihasilkan adalah cahaya yang sedang.



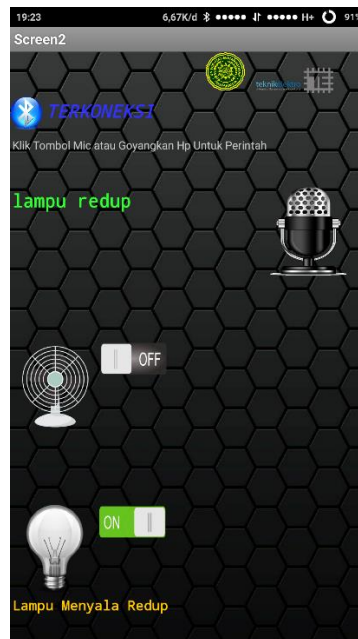
Gambar 4.27 Tampilan aplikasi pada saat lampu sedang



Gambar 4.28 Indikator LED pada saat lampu sedang

### 3. Pemberian perintah lampu redup

Pemberian perintah lampu redup merupakan perintah untuk memberikan *output* tegangan rendah pada beban lampu untuk mendapatkan hasil cahaya lampu yang redup.



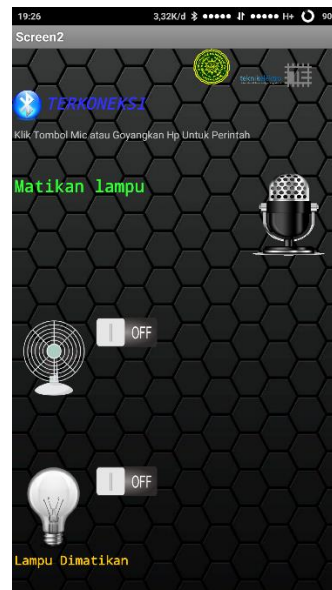
Gambar 4.29 Tampilan aplikasi pada saat lampu redup



Gambar 4.30 Indikator LED pada saat lampu redup

#### 4. Pemberian perintah lampu dimatikan

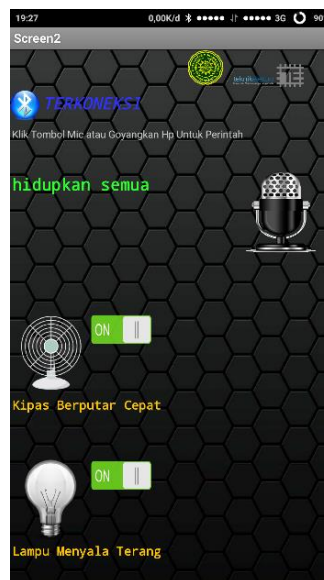
Pemberian perintah matikan lampu digunakan untuk memutus tegangan *output* pada beban lampu sehingga lampu tidak menyala.



Gambar 4.31 Tampilan aplikasi pada saat lampu dimatikan

c. Pengujian kinerja alat pada beban kipas dan lampu

Dengan perintah hidupkan semua maka beban kipas dan lampu akan aktif pada kondisi maksimal semua. Kemudian pada perintah matikan semua maka kedua beban akan berubah pada kondisi *off*.



Gambar 4.32 Tampilan aplikasi pada saat perintah hidupkan semua



Gambar 4.33 Tampilan aplikasi pada saat perintah matikan semua

Pengujian kinerja selanjutnya adalah pengujian jarak jangkauan pengendalian alat. Dengan menggunakan koneksi *bluetooth* maka pengendalian alat ini memiliki keterbatasan jarak jangkauan. Penggunaan koneksi *bluetooth* digunakan karena kemudahan dalam penggunaan dan mudah dalam penggunaan modul *bluetooth* yang digunakan.

Jarak jangkauan koneksi *bluetooth* pada perangkat-perangkat yang ada dipasaran adalah berkisar pada jarak 10 m . Jarak jangkauan koneksi *bluetooth* sangat berpengaruh pada area yang digunakan dalam perangkat ini. Apabila area tersebut terhalang oleh benda ataupun tembok maka akan mempengaruhi jarak jangkauan *bluetooth* tersebut. Dalam pengujian alat ini tida terhalang oleh tembok.

Adapun data pengukuran jarak jangkauan alat ini dapat dilihat pada tabel berikut :

Tabel 4.7 Kemampuan jarak koneksi *Bluetooth*

| <b>JARAK</b> | <b>KONEKSI</b>   |
|--------------|------------------|
| 1 Meter      | TERKONEKSI       |
| 2 Meter      | TERKONEKSI       |
| 3 Meter      | TERKONEKSI       |
| 4 Meter      | TERKONEKSI       |
| 5 Meter      | TERKONEKSI       |
| 6 Meter      | TERKONEKSI       |
| 7 Meter      | TERKONEKSI       |
| 8 Meter      | TERKONEKSI       |
| 9 Meter      | TERKONEKSI       |
| 10 Meter     | TERKONEKSI       |
| 11 Meter     | TIDAK TERKONEKSI |

Dibawah ini merupakan tabel data percobaan dari beberapa orang yang memberikan perintah suara. Adapun hasilnya adalah dapat dilihat pada tabel dibawah ini :

Tabel 4.8 Daftar nama penguji aplikasi

| No | Nama      | Pria/Wanita | Keterangan        |
|----|-----------|-------------|-------------------|
| 1  | Popi      | Wanita      | <b>Terdeteksi</b> |
| 2  | Arya      | Pria        | <b>Terdeteksi</b> |
| 3  | Danardono | Pria        | <b>Terdeteksi</b> |
| 4  | Ibu Ana   | Wanita      | <b>Terdeteksi</b> |
| 5  | Andi      | Pria        | <b>Terdeteksi</b> |

Dari data pada tabel diatas dilakukan pengujian pada setiap sample penguji dengan memberikan perintah suara sesuai perintah yang ada. Dari hasil data yang didapatkan apabila orang tersebut memberikan perintah dengan jelas dan bahasa yang sesuai maka dapat terdeteksi semua.

Dari hasil yang didapatkan sesuai data pada tabel diatas maka dapat diambil kesimpulan bahwa aplikasi yang telah penulis buat yaitu perintah suara dapat digunakan semua orang baik pria maupun wanita dengan syarat pengucapan perintah

harus dengan jelas dan bahasa yang digunakan harus sesuai dengan perintah yang digunakan dalam aplikasi tersebut.

Selanjutnya adalah data pengujian kehandalan pembacaan, yaitu sejauh mana aplikasi yang terpasang pada *smartphone* dapat menerima masukan perintah dari pengguna. Adapun hasil dari pengujian dapat dilihat pada tabel dibawah ini :

Tabel 4.9 Data kehandalan aplikasi

| NO | Jarak pembicara dengan Smartphone | Keterangan              |
|----|-----------------------------------|-------------------------|
| 1  | 10 cm                             | <b>Terdeteksi</b>       |
| 2  | 30 cm                             | <b>Terdeteksi</b>       |
| 3  | 50 cm                             | <b>Terdeteksi</b>       |
| 4  | 70 cm                             | <b>Terdeteksi</b>       |
| 5  | 100 cm                            | <b>Tidak Terdeteksi</b> |

Pengujian dilakukan dengan pengukuran pemberi suara yaitu pada pengukuran bibir sebagai sumber suara dengan *smartphone* sebagai penangkap suara melalui *microfon*. Dari hasil data pada tabel diatas dapat disimpulkan bahwa jarak efektif yang dapat digunakan untuk memberikan suara dalah pada jarak dibawah 1 meter dengan



suara yang sedang, apabila dengan suara berteriak dimungkinkan bisa mencapai jarak yang lebih jauh.

Data pengujian selanjutnya merupakan data pengujian kecepatan respon dari aplikasi untuk mengontrol beban. Adapun data yang didapatkan dapat dilihat pada tabel dibawah ini :

Tabel 4.10 Data Kecepatan Respon Aplikasi

| <b>Pengontrolan<br/>Dengan Alat perintah Suara</b> | <b>Pengontrolan<br/>Dengan Saklar Konvensional</b> |
|--|--|
| 2 Detik  | Jarak 3m = 3 detik                                 |
|  | Jarak 5m = 6 detik                                 |
|  | Jarak 10m = 8 detik                                |

Pengujian dilakukan dengan membandingkan kecepatan yang didapatkan dari penggunaan alat perinth suara dan saklar konvensional. Dari data yang didapatkan sesuai dengan tabel diatas dapat ditarik kesimpulan bahwa dengan menggunakan alat perintah suara ini pengontrolan beban akan jauh lebih cepat dan lebih mudah, karena hanya dengan memberikan perintah suara.

Pengujian jarak menggunakan saklar konvensional dilakukan pada posisi berjalan dari titik awal menuju lokasi saklar dalam posisi berdiri, apabila posisi awal

pengguna adalah pada posisi tertidur atau dalam posisi duduk maka untuk menjangkau saklar akan lebih memakan banyak waktu untuk pengontrolan bebannya.