

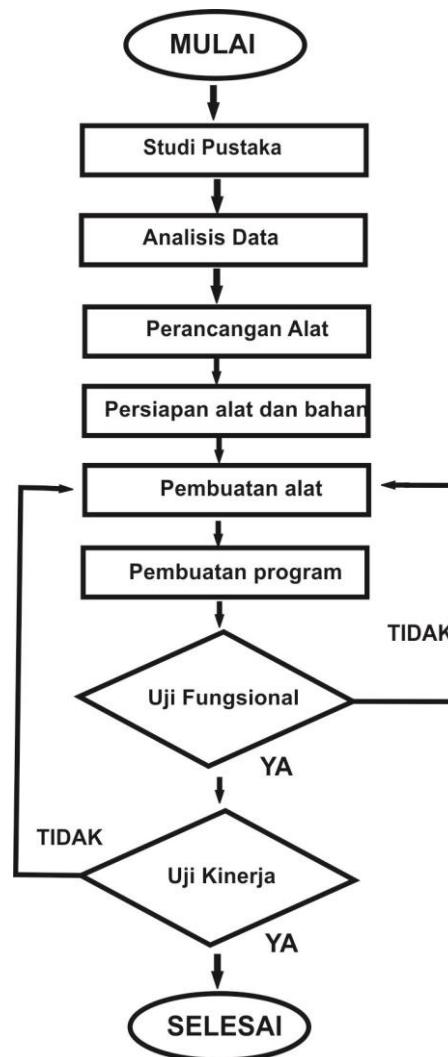
BAB III

METODOLOGI PERANCANGAN

3.1 Diagram Alir Penelitian

Pada penelitian ini di perlukan diagram alir yang digunakan untuk mengetahui langkah-langkah yang perlu dilakukan untuk menyelesaikan alat ini.

Diagram alir digambarkan pada gambar berikut :



3.1 Gambar Diagram Alir Penelitian

3.1.1 Penjelasan diagram alir penelitian

a. Mulai

Pada proses ini penulis memulai persiapan untuk mengerjakan penelitian dan pembuatan Sistem Pengontrol Kecepatan Kipas Dan Kecerahan Lampu Dengan Perintah Suara Berbasis *Android*.

b. Studi Pustaka

Pada tahap ini dilakukan pengumpulan data-data dan informasi yang berkaitan dan akan membantu dalam proses penelitian. Pengumpulan data-data dilakukan melalui media internet dan buku.

c. Analisis Data

Data-data dan informasi yang telah didapatkan kemudian dilakukan analisis untuk menentukan bentuk, bahan dan komponen yang akan digunakan dalam penelitian ini.

d. Perancangan Alat

Pada penelitian ini dimulai dengan melakukan perancangan alat. Pada tahap ini dilakukan perancangan alat berupa bentuk, penggunaan komponen dan tata letak dari komponen tersebut.

e. Persiapan Alat dan Bahan

Pada tahap ini penulis mempersiapkan bahan-bahan yang nantinya akan digunakan dalam pembuatan alat. Dengan adanya persiapan bahan proses pembuatan alat bisa segera dimulai. Penggunaan alat-alat yang digunakan dalam pembuatan alat ini juga disiapkan untuk mempermudah ataupun mempercepat proses perancangan dan pembuatan.

f. Pembuatan Alat

Pada proses ini merupakan proses utama dari pembuatan alat yang berupa *hardware* dengan menyesuaikan dengan tahapan perencanaan. Pembuatan *hardware* dilakukan secara berurutan dari pembuatan PCB, pemasangan komponen atau modul, *wiring* dan pembuatan *box*.

g. Pembuatan Program

Pada tahap ini merupakan tahap untuk pembuatan program yang nantinya akan dimasukkan pada *hardware* yang telah selesai dibuat. Pemrograman yang dilakukan adalah pemrograman untuk *arduino* yaitu dengan bahasa C. *Arduino* akan ditempatkan didalam *box* sebagai pengontrol beban secara mikrokontroler. Selain pemrograman untuk *arduino*, dilakukan juga pemrograman untuk pembuatan aplikasi *Android* sebagai perangkat pengendali dari alat ini.

h. Uji Fungsional

Pada tahap ini dilakukan pengujian alat secara fungsional. Pengujian dilakukan secara per bagian dari alat untuk mengetahui apakah alat ini dapat berfungsi dengan baik per bagian dari *hardware* ataupun dari program yang diberikan. Pada tahap ini apabila terjadi *error* ataupun ketidaksesuaian secara fungsional maka akan dilakukan pengecekan kembali baik dari segi *hardware* ataupun secara *software*.

i. Uji Kinerja

Pada proses ini dilakukan pengujian kinerja alat dari segi kemampuan, ketahanan alat dan daya jangkau dari koneksi *bluetooth* yang digunakan pada

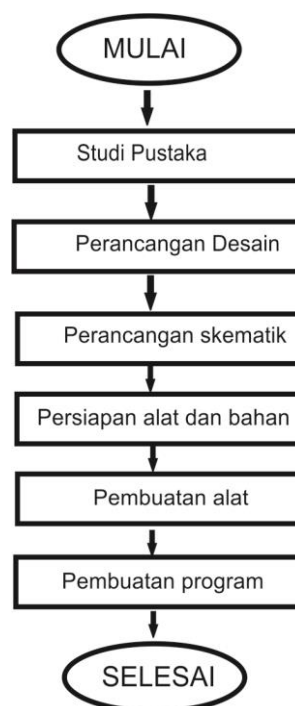
alat ini. Pada tahap ini apabila terjadi *error* ataupun ketidaksesuaian kinerja maka akan dilakukan pengecekan kembali baik dari segi *hardware* ataupun secara *software*.

j. Selesai

Dalam hal ini penulis telah menyelesaikan tugas akhir baik dalam pembuatan alat , pengujian alat dan perbaikan alat dan selanjutnya alat ini sudah dapat diaplikasikan dengan baik.

3.2 Perancangan Alat

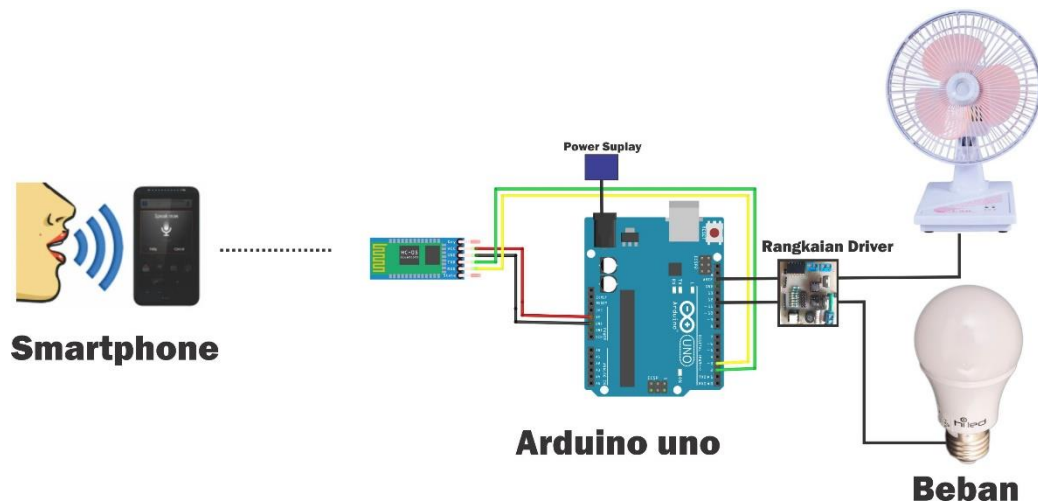
Pada tahap perancangan alat ini terdapat langkah-langkah yang penulis lakukan dengan mengacu pada diagram alir proses perancangan alat sesuai gambar dibawah ini :



Gambar 3.2 Diagram Alir Perancangan Alat

Pada perancangan alat ini terdapat dua bagian utama yaitu *smartphone* sebagai perangkat pengirim data dari perintah suara dan rangkaian *arduino* sebagai

penerima data dan pengatur beban berupa kipas dan lampu. Adapun blok diagram dari perancangan alat ini adalah sebagai berikut :



Gambar 3.3 Blok diagram alat

Pada gambar blok diatas terdapat tiga bagian utama dalam pembuatan alat pengaturan kecepatan kipas dan kecerahan lampu dengan perintah suara melalui *smartphone*. Tiga bagian utama tersebut adalah sebagai berikut :

a. *Smartphone (Input)*:

Smartphone android merupakan perangkat yang digunakan sebagai *input* dari alat ini. *Smartphone android* berfungsi sebagai perangkat yang mengubah perintah suara yang bersumber dari suara manusia menjadi data digital yang kemudian akan dikirimkan ke bagian proses yaitu *arduino uno R3* melalui koneksi *bluetooth*. Pada penggunaan *smartphone* ini memanfaatkan fasilitas *voice recognition* yaitu fasilitas yang dapat digunakan untuk mengkonversi suara manusia yang berupa kata-kata menjadi data teks digital. Dengan penggunaan *smartphone android* yang

mengijinkan siapapun untuk membuat *software android* maka penulis dapat membuat aplikasi *android* yang memiliki fungsi mengaktifkan *voice recognition* dan data hasil konversi dapat langsung dikirim melalui koneksi *bluetooth*. Aplikasi yang digunakan dalam pembuatan *software android* tersebut adalah App Inventor yang memiliki fitur-fitur khusus untuk pembuatan *software android*.

b. *Arduino* Uno R3(Proses) :

Arduino uno R3 dengan bahasa pemrograman bahasa C merupakan perangkat ataupun *hardware* yang memiliki fungsi utama sebagai pemrosesan data. Dengan tambahan modul *Bluetooth* HC-05 *arduino* uno R3 dapat berkomunikasi secara serial dengan perangkat *input* yang berupa *smartphone android*. Pembuatan program pada *arduino* uno R3 menggunakan *software Arduino IDE* dengan bahasa pemrograman C. Dengan pemberian program pada *arduino* uno R3 ini maka *arduino* dapat berfungsi sebagai penerima data *input* dari *smartphone* melalui komunikasi serial dengan koneksi *bluetooth* kemudian data akan diproses. Hasil data yang didapatkan akan disesuaikan dengan program yang diberikan untuk pengaturan *output* ke rangkaian beban yang berupa kipas angin 220v AC dan lampu led 220v AC.

c. Beban Kipas dan Lampu LED 220v AC (*Output*) :

Pada blok ini merupakan bagian dari output beban yaitu berupa kipas angin 220v AC dan lampu led 220v AC. Pengontrolan beban ini dikendalikan melalui port *output arduino* uno R3 dengan tambahan rangkaian optocoupler MOC 3021 dan triac BTA 12 sebagai penghubung dari rangkaian mikrokontroler dengan tegangan DC 5v dan beban yang bekerja pada tegangan AC 220V. Penggunaan

rangkaian pengendali tegangan Ac yang berupa triac juga dapat berfungsi untuk pengaturan tegangan menuju beban kipas dan lampu led, sehingga dapat diatur tingkat kecepatan kipas dan tingkat kecerahan lampu led tersebut. Namun dalam penggunaan lampu led 220v AC dengan spesifikasi yang dimmable pada rangkaian kontrolnya perlu ditambahkan rangkaian *Zero Crossing* berupa ic 4n25 dan *diode bridge* untuk mendapatkan hasil pengaturan kecerahan lampu yang maksimal.

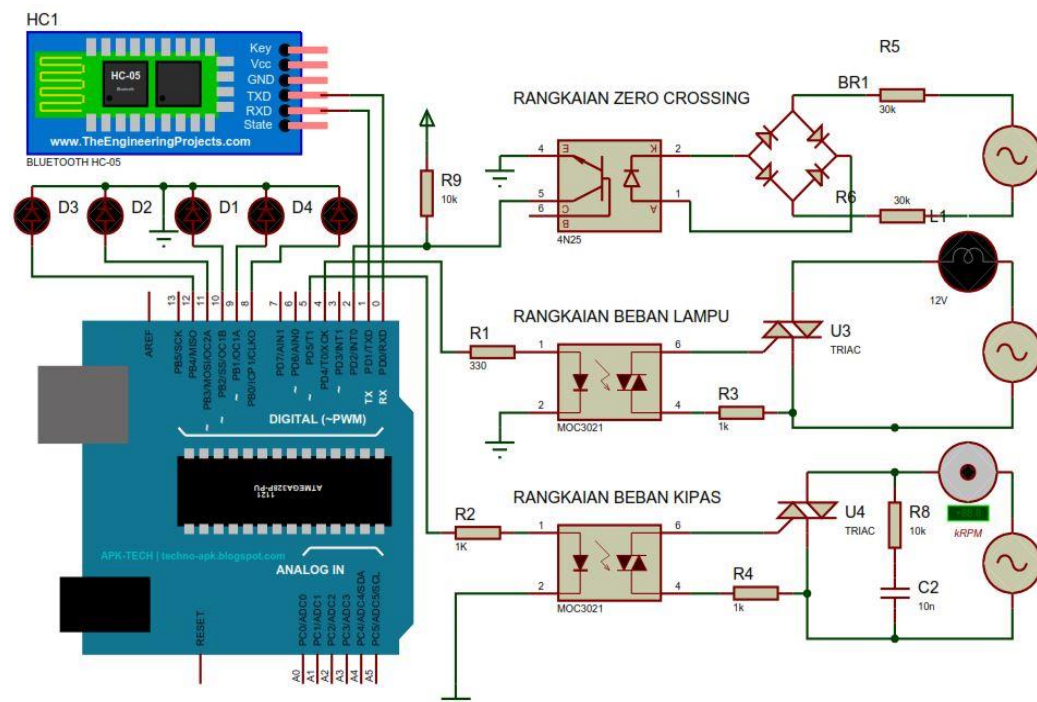
3.2.1 Perancangan *Hardware*

a. Perancangan *Arduino* uno R3

Dalam perancangan sistem pemrosesan digital menggunakan mikrokontroler *arduino* uno perlu dilakukan persiapan fungsi-fungsi pin pada *arduino* untuk dapat digunakan sesuai dengan yang kehendaki. Penggunaan *Arduino* Uno R3 tidak perlu membuat sistem minimumnya dikarenakan modul ini sudah merupakan board yang dapat langsung digunakan tinggal menambahkan sumber tegangan dan memberikan program di dalamnya. Fungsi utama *Arduino* Uno R3 pada alat ini adalah sebagai pemrosesan data utama dari *input* untuk mengontrol rangkaian *output* beban berupa kipas dan lampu led dengan tegangan AC 220v. Tahapan dalam perancangan yang dilakukan adalah sebagai berikut :

Perancangan diagram skematik menggunakan *software* PROTEUS ISIS. Pada tahap ini yang dilakukan adalah menyiapkan seluruh komponen yang akan digunakan dalam perancangan skematik pada *software* PROTEUS ISIS. Komponen-komponen yang sudah didapatkan kemudian dilakukan integrasi antar komponen sesuai dengan tahapan perencanaan untuk dapat membuat skematik

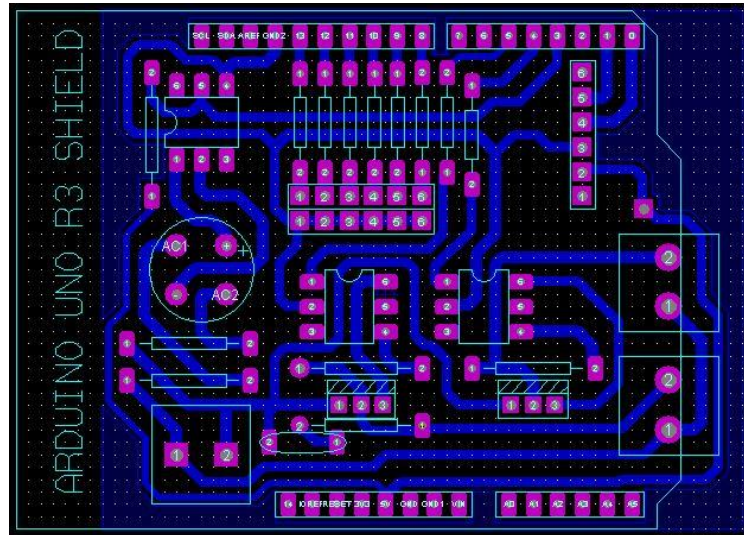
rangkaian alat secara baik. Adapun rancangan skematik rangkaian *Arduino* dan rangkaian pengendali beban adalah sebagai berikut :



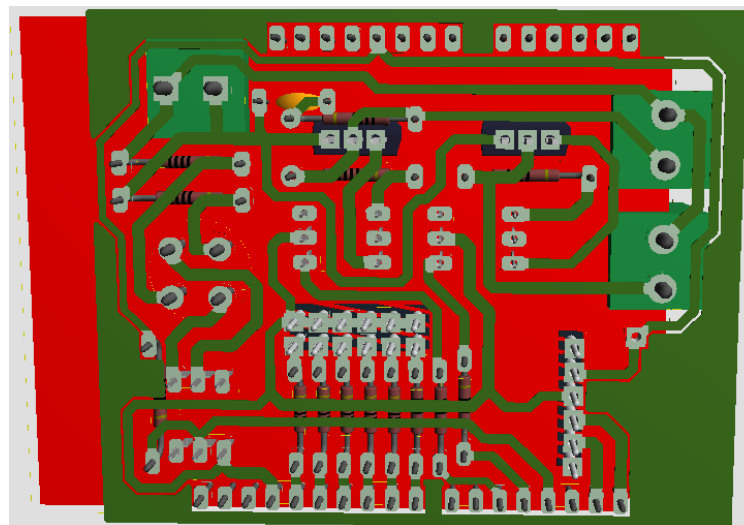
Gambar 3.4 Skematik *Arduino* Uno R3

Perancangan layout PCB dengan menggunakan *software* PROTEUS ARES. Perancangan *layout* PCB bertujuan untuk mengatur tata letak komponen yang telah dipersiapkan dan dibuat dalam skematik untuk di integrasikan didalam papan PCB yang sebenarnya. *Software* PROTEUS ARES digunakan untu membuat layout PCB yang akan dicetak kedalam papan PCB menggunakan *art paper* kemudian dilarutkan.

Adapun gambar *layout* PCB menggunakan *software* PROTEUS ARES adalah sebagai berikut :



Gambar 3.5 Layout PCB



Gambar 3.6 Tampilan Layout 3D

Komponen utama pengendali diatas adalah *Arduino* Uno R3 dan komponen tambahannya adalah modul *Bluetooth* HC-05 dan rangkaian pengontrol beban berupa rangkaian *zero crossing detector* dan ic MOC 3021 dengan triac. Dengan

pembuatan jalur rangkaian pada PCB ini bertujuan untuk mempermudah dalam pemasangan komponen yang ada.

Adapun pin I/O yang digunakan pada *Arduino* Uno R3 dapat dilihat pada tabel berikut :

Tabel 3.1 Penggunaan Pin *Arduino*

Pin	Fungsi	Keterangan
0	RX	Penerima komunikasi serial HC-05
1	TX	Pengirim komunikasi serial HC-05
3	INT1	Masukan interrupt dari <i>output zero crossing</i>
4	OUT	<i>Output</i> Pengendali optoisolator beban Lampu
5	PWM	<i>Output</i> Pengendali optoisolator beban Kipas
8	OUT	<i>Output</i> LED indicator
9	OUT	<i>Output</i> LED indicator
10	OUT	<i>Output</i> LED indicator
11	OUT	<i>Output</i> LED indicator
12	OUT	<i>Output</i> LED indicator
13	OUT	<i>Output</i> LED indicator

Pada pin RX *Arduino* Uno R3 tersambung dengan pin Rx modul *Bluetooth* HC-05. Apabila *Bluetooth* HC-05 sudah terkoneksi maka jika ada data yang dikirimkan dari *smartphone* melalui *bluetooth* data tersebut akan masuk ke *arduino* dan dapat terbaca untuk dieksekusi sesuai dengan perintah pada program. Pada bagian *output Arduino* Uno terdapat rangkaian pengontrol beban yaitu pada pin 5 yaitu *output* pwm yang terhubung dengan ic MOC 3021 dan TRIAC BTA 12 untuk terhubung ke beban kipas AC 220V dan dapat dilakukam pengontrolan kecepatan.

Pada rangkaian optoisolator pengendali beban kipas ditambahkan resistor 1K ohm dan kapasitor 100nF 400v sebagai rangkaian snubber untuk menstabilkan pada saat pengontrolan beban kipas angin. Kemudian pada pengontrolan beban lampu led terdapat rangkaian tambahan *zero crossing* berupa *diode bridge* , resistor 30k dan optotransistor dengan *output* masuk ke pin interrupt 1 pada *arduino*.

Rangkaian *zero crossing* ini berfungsi untuk mengetahui titik persilangan pada tegangan AC bolak balik untuk pengontrolan lampu led AC 220V. Pengontrolan beban lampu led juga ditambahkan rangkaian optoisolator dan TRIAC BTA12 untuk dapat berfungsi mengontrol kecerahan lampu atau sebagai dimmer. Pada pin 8 sampai pin 13 digunakan untuk *output* led indikator pada saat kipas berputar cepat, lambat dan lampu pada saat menyala redup, sedang dan terang.

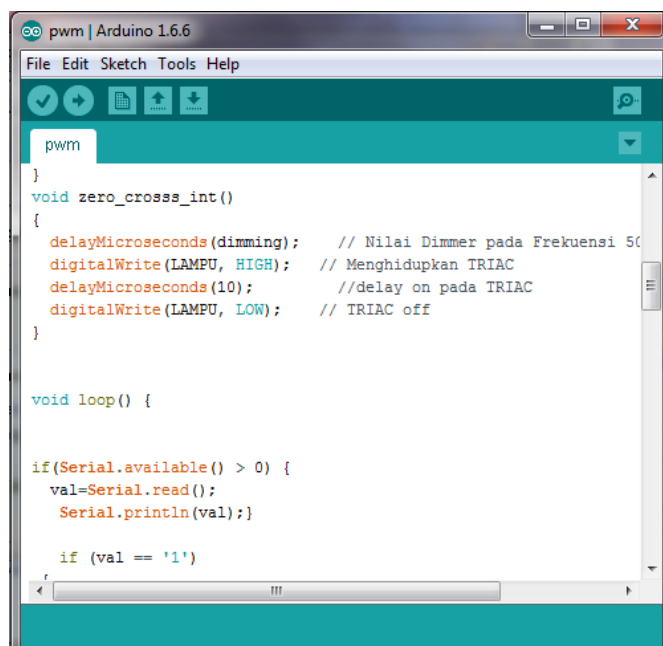
3.2.2 Perancangan Program dan Aplikasi

Pada perancangan pemrograman terbagi menjadi dua bagian perancangan. Pemrograman digunakan untuk memberikan perintah pada perangkat ataupun mikrokontroler untuk mengeksekusi sesuai perintah program.

Adapun pemrograman yang penulis lakukan adalah sebagai berikut :

a. Pemrograman *Arduino Uno R3*

Pada pemrograman *Arduino Uno R3* menggunakan *software Arduino IDE*. Dengan *software* tersebut dapat penulis melakukan pemrograman melalui komputer dengan menggunakan library yang ada dan perintah-perintah pemrograman bahasa C pada *Arduino IDE*.



```

pwm | Arduino 1.6.6
File Edit Sketch Tools Help
pwm
}
void zero_crossss_int()
{
  delayMicroseconds(dimming); // Nilai Dimmer pada Frekuensi 50
  digitalWrite(LAMPU, HIGH); // Menghidupkan TRIAC
  delayMicroseconds(10); //delay on pada TRIAC
  digitalWrite(LAMPU, LOW); // TRIAC off
}

void loop() {

if(Serial.available() > 0) {
  val=Serial.read();
  Serial.println(val);}

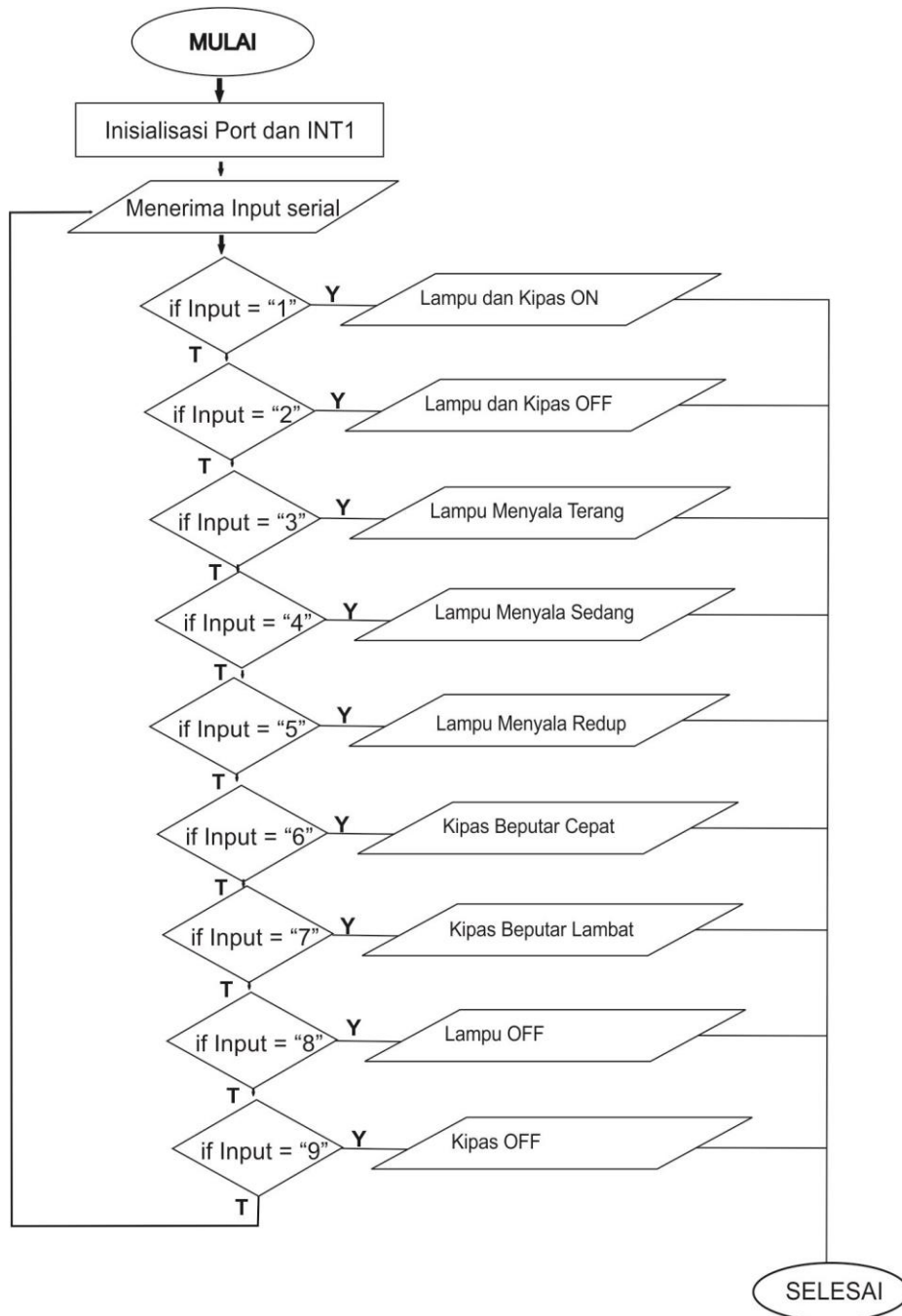
  if (val == '1')
  
```

Gambar 3.7 Program *Arduino IDE*

Pemrograman *arduino* menggunakan bahasa pemrograman C. Dengan menggunakan *software Arduino IDE* terdapat fungsi untuk menuliskan library yang dapat digunakan untuk membantu dalam penulisan program. Pemrograman *arduino* dilakukan dengan cara menentukan pin atau fungsi yang akan digunakan ,kemudian membuat pemanggilan dan inisialisasi *port* pada *arduino*. Apabila port sudah disiapkan untuk dimasukan didalam program maka langkah selanjutnya adalah melakukan penulisan program dengan mengacu pada *flowchart* yang telah dibuat

dan dengan menyesuaikan perintah-perintah yang akan digunakan pada *software arduino IDE*.

Adapun diagram alir dari pemrograman pada *Arduino Uno* sebagai perangkat pemroses data adalah sebagai berikut :



Gambar 3.8 Flowchart Program Arduino Uno

Adapun untuk list program yang penulis buat untuk program pada Arduino Uno adalah sebagai berikut :

```

int LAMPU = 4; // Output ke Opto Triac beban lampu
int dimming; // Nilai dimming zero crossing
char val;
void setup() {
  Serial.begin(9600); // Pengaturan Baudrate
  pinMode(4,OUTPUT); // output TRIAC pada beban lampu
  pinMode(5,OUTPUT); // output PWM TRIAC pada beban Kipas
  pinMode(9,OUTPUT); // indikator LED
  pinMode(10,OUTPUT); // indikator LED
  pinMode(11,OUTPUT); // indikator LED
  pinMode(12,OUTPUT); // indikator LED
  pinMode(13,OUTPUT); // indikator LED

  digitalWrite(9,HIGH);
  digitalWrite(11,HIGH);
  digitalWrite(10,HIGH);
  digitalWrite(12,HIGH);
  digitalWrite(13,HIGH);
  delay(1000);
  digitalWrite(9,LOW);
  digitalWrite(11,LOW);
  digitalWrite(10,LOW);
  digitalWrite(12,LOW);
  digitalWrite(13,LOW);}
void zero_crosss_int() // Fungsi Zero Crossing
{
  delayMicroseconds(dimming); // Nilai Dimmer pada Frekuensi 50HZ (1-10 ms)
  digitalWrite(LAMPU, HIGH); // Menghidupkan TRIAC
  delayMicroseconds(10); //delay on pada TRIAC
  digitalWrite(LAMPU, LOW); // TRIAC off
}
void loop() {
  if(Serial.available() > 0) { // Pembacaan data serial
    val=Serial.read();
    Serial.println(val);}

  if (val == '1') // Perintah jika nilai input "1"

```

```
{
analogWrite(5,255);
attachInterrupt(1, zero_crosss_int,RISING);
dimming=1200;
digitalWrite(9,HIGH);
digitalWrite(11,HIGH);
digitalWrite(10,LOW);
digitalWrite(12,LOW);
digitalWrite(13,LOW);
}
else if (val == '2')    // Perintah jika nilai input "2"
{
analogWrite(5,0);
attachInterrupt(1, zero_crosss_int,RISING);
dimming=0;
digitalWrite(9,LOW);
digitalWrite(10,LOW);
digitalWrite(11,LOW);
digitalWrite(12,LOW);
digitalWrite(13,LOW);
}

else if (val == '3')    // Perintah jika nilai input "3"
{
attachInterrupt(1, zero_crosss_int,RISING);
dimming=1200;
digitalWrite(11,HIGH);
digitalWrite(9,LOW);
digitalWrite(10,LOW);
digitalWrite(12,LOW);
digitalWrite(13,LOW);
}

else if (val == '4')    // Perintah jika nilai input "4"
{
attachInterrupt(1, zero_crosss_int,RISING);
dimming=7000;
digitalWrite(12,HIGH);
```

```
digitalWrite(11,LOW);
digitalWrite(13,LOW);
}

else if (val == '5')      // Perintah jika nilai input "5"
{
attachInterrupt(1, zero_crosss_int,RISING);
dimming=8500;
digitalWrite(13,HIGH);
digitalWrite(11,LOW);
digitalWrite(12,LOW);
}

else if (val == '6')      // Perintah jika nilai input "6"
{
analogWrite(5,255);
digitalWrite(9,HIGH);
digitalWrite(10,LOW);
}

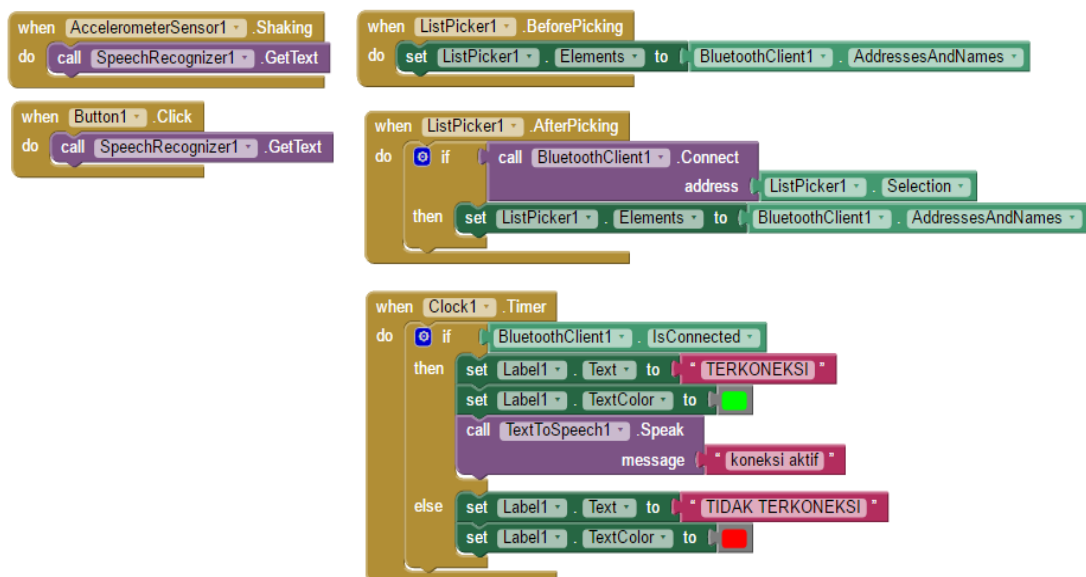
else if (val == '7')      // Perintah jika nilai input "7"
{
analogWrite(5,4);
digitalWrite(10,HIGH);
digitalWrite(9,LOW);
}

else if (val == '8')      // Perintah jika nilai input "8"
{
attachInterrupt(1, zero_crosss_int,RISING);
dimming=0;
digitalWrite(11,LOW);
digitalWrite(12,LOW);
digitalWrite(13,LOW);}
else if (val == '9')      // Perintah jika nilai input "9"
{analogWrite(5,0);
digitalWrite(9,LOW);
digitalWrite(10,LOW);}

}
```


b. Pemrograman aplikasi perintah suara untuk *Smartphone*

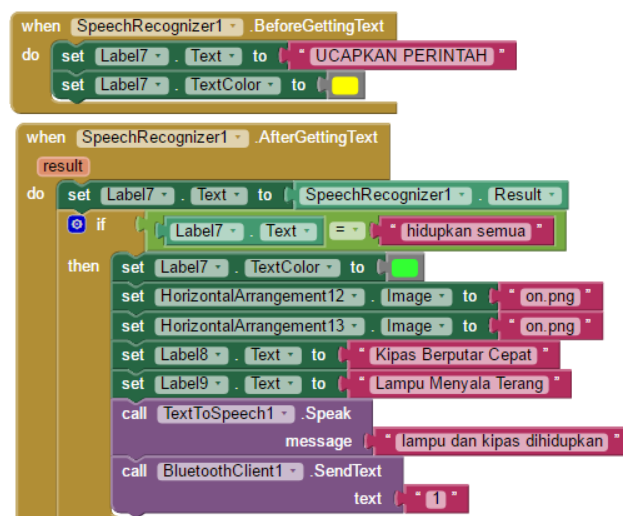
Pada perangkat *Smartphone android* yang akan digunakan sebagai media pengirim perintah suara harus sudah terinstal aplikasi perintah suara. Dalam pembuatan aplikasi dan programnya penulis menggunakan *software* App Inventor dengan menggunakan fungsi *voice recognition* sebagai pemrosesan data dari suara manusia menjadi *text* yang kemudian akan dikirimkan melalui koneksi *bluetooth* ke *arduino*. Adapun program pada APP Inventor tersebut adalah sebagai berikut :



Gambar 3.9 Program koneksi *Bluetooth* App Inventor

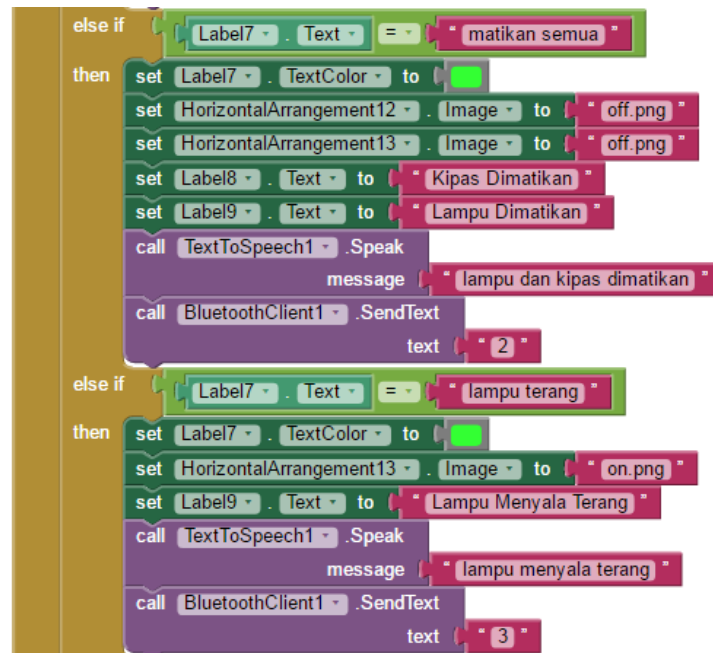
Pada program diatas merupakan list program yang digunakan untuk pengaktifan fungsi *bluetooth* pada *smartphone*. Nantinya didalam aplikasi yang dibuat pengguna dapat memilih nama *bluetooth* yang akan dikoneksikan dari list data *bluetooth* yang disediakan. Fungsi clock timer digunakan untuk memberikan indicator koneksi *bluetooth* dengan teks berwarna apabila sudah terkoneksi maka

akan muncul teks bertuliskan koneksi aktif berwarna hijau. Terdapat pengaktifan fungsi *speech recognition* dan sensor *accelerometer* untuk mengaktifkan fungsi *input* suara. Apabila sensor *accelerometer* diaktifkan dengan cara menggoyangkan perangkat *smartphone*, maka akan muncul fungsi *speech recognition* untuk diberikan *input* suara perintah.



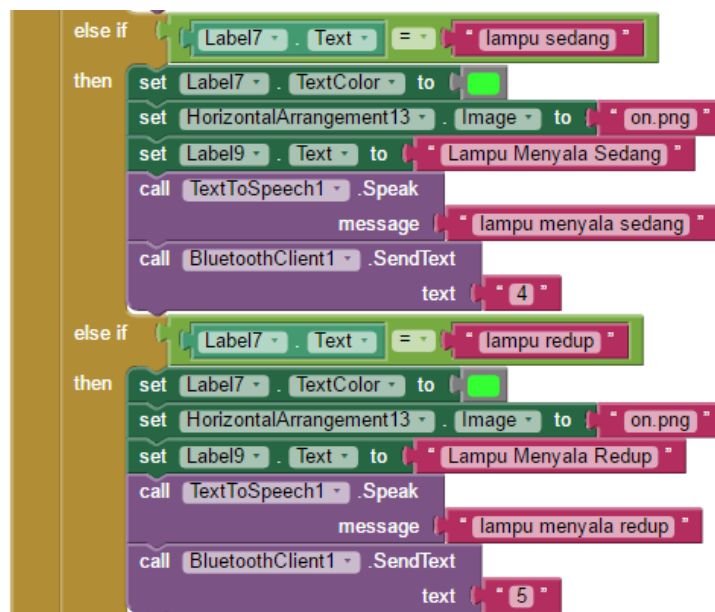
Gambar 3.10 Program *Speech Recognition* App Inventor

Pada list program selanjutnya adalah program data perintah yang akan dieksekusi apabila hasil dari *speech recognition* sesuai dengan data pada program ini. Fungsi *speech recognition* akan mengubah sumber perintah manusia menjadi teks yang kemudian teks tersebut akan dibandingkan, apabila teks yang dibandingkan sesuai dengan data yang ada pada program maka *smartphone* akan mengirimkan angka melalui koneksi *bluetooth* menuju ke *arduino* untuk selanjutnya diproses untuk menggerakkan *output* beban.

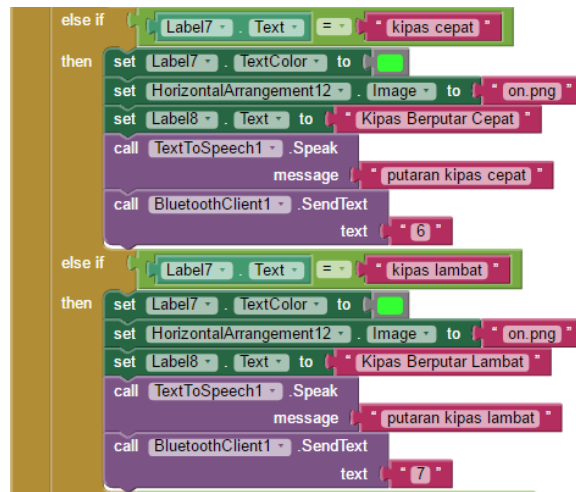


Gambar 3.11 Program perintah 1-3 App Inventor

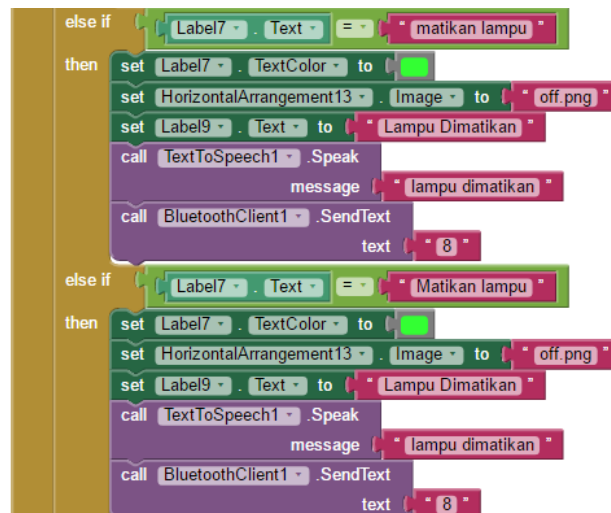
Pada program diatas adalah program perintah yang didapatkan dari input google voice dengan pengaktifan speech recognition.



Gambar 3.12 Program perintah 4-5 App Inventor



Gambar 3.13 Program perintah 6-7 App Inventor



Gambar 3.14 Program perintah 7-8 App Inventor



Gambar 3.15 Program perintah 9 App Inventor

3.2.3 Prinsip kerja alat

Prinsip kerja dari alat ini adalah dengan mengkoneksikan *smartphone* dengan *Bluetooth* HC-05 kemudian memberikan masukan berupa perintah suara pada *smartphone android* yang sudah di *install* aplikasi perintah suara. Apabila perintah yang diucapkan sesuai dengan perintah pada aplikasi maka akan dieksekusi dengan mengirimkan data berupa angka melalui koneksi *bluetooth* ke *Arduino* Uno. Sebagai contohnya diberikan perintah “nyalakan semua” maka *smartphone* akan mengirimkan nilai 1, data tersebut akan diproses oleh *Arduino* untuk melakukan pengontrolan beban berupa kipas dan lampu led 220v AC dengan memberikan *output* ke rangkaian pengontrol beban pada Triac untuk menghidupkan kipas dan lampu sesuai perintah yang diberikan.

Apabila perintah yang diucapkan tidak sesuai dengan daftar perintah yang ada pada aplikasi di *smartphone* maka tidak ada yang dieksekusi pada *Arduino* atau tidak akan berpengaruh pada beban yang dikontrol.

3.3 Alat dan Bahan

Pembuatan alat meliputi realisasi dari seluruh rancangan rangkaian yang telah dibuat. Dimulai dari persiapan alat dan bahan yang akan digunakan dan tahapan pembuatan alat.

3.3.1 Bahan yang dibutuhkan

- a. Kipas 220V AC
- b. Lampu LED 220V AC
- c. *BOX*
- d. *Arduino* Uno R3

- e. Modul *Bluetooth* HC-05
- f. Triac BTA 12
- g. Ic MOC 3021
- h. IC 4N25
- i. Dioda Bridge
- j. Resistor
- k. Kapasitor
- l. Papan PCB
- m. LED
- n. Pin Header
- o. Kabel
- p. Adaptor 220v AC to 5v dc

3.3.2 Peralatan yang dibutuhkan

- a. Solder
- b. Timah
- c. Penggaris
- d. Multimeter
- e. Obeng
- f. Bor listrik

3.3.3 *Software* yang dibutuhkan

- a. *Arduino* IDE
- b. App Inventor
- c. Proteus 7

3.4 Realisasi

Pada tahap realisasi alat dimulai dengan mengimplementasikan rancangan komponen dari skema yang telah dibuat di proteus. Pembuatan jalur pada papan PCB menggunakan teknik sablon menggunakan *art paper* yang telah di print menggunakan tinta laser. Kertas *art paper* tersebut kemudian di setrika pada permukaan tembaga pada papan PCB dengan tujuan menempelkan jalur yang sudah tercetak pada *art paper*. Setelah jalur rangkaian sudah tercetak atau menempel pada papan PCB maka tahapan selanjutnya adalah melakukan pelarutan PCB menggunakan cairan FeCL3.

Larutan FeCL3 berfungsi sebagai pelarut dari tembaga yang tidak tertutup oleh jalur yang didapatkan dari *art paper* agar terbentuk jalur tembaga yang diinginkan. Pelarutan dilakukan sekitar 15 menit dengan cara menggoyangkan wadah dari pelarut untuk mendapatkan hasil pelarutan yang cepat. Setelah tembaga yang tidak tertutup oleh jalur sudah hilang atau terlarut maka langkah selanjutnya adalah membersihkan sisa *art paper* dan sisa tinta toner yang masih menempel pada papan PCB. Setelah proses pelarutan selesai maka dilakukan proses penyolderan komponen sesuai dengan layout yang telah dibuat pada *software* Proteus ARES.



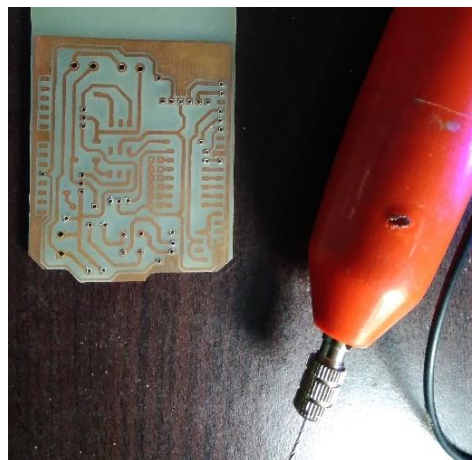
Gambar 3.16 Proses pelarutan PCB

Pada gambar dibawah ini adalah pengambilan gambar pada saat proses pencetakan jalur dengan teknik pemindahan jalur ke PCB dengan mesin press. Dengan setting panas dan kecepatan putar yang sesuai maka hasil jalur akan menempel di papan PCB dengan baik.



Gambar 3.17 Proses Sablon PCB

Gambar dibawah ini merupakan gambar pada saat proses pengeboran papan PCB. Pengeboran dilakukan dengan tujuan untuk tempat pemasangan kaki-kaki komponen pada papan PCB yang sudah terbuat jalur rangkaiannya.



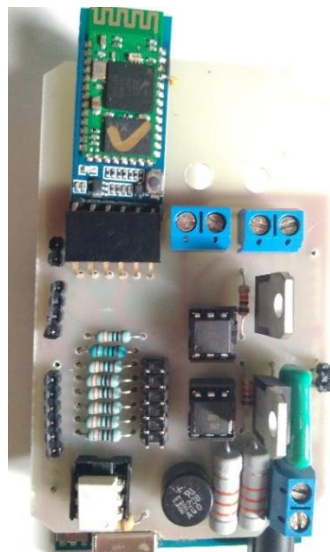
Gambar 3.18 Proses Pengeboran PCB

Pada gambar dibawah ini merupakan proses penyolderan kaki-kaki komponen yang sudah terpasang pada papan PCB. Penyolderan dilakukan dengan tujuan untuk memperkuat sambungan antara kaki komponen dengan jalur PCB dan memastikan dapat terkoneksi dan dapat dialiri arus dengan baik.



Gambar 3.19 Proses Penyolderan

Pada gambar dibawah ini merupakan gambar papan PCB yang sudah selesai dalam tahap pemasangan komponen untuk selanjutnya dilakukan pengujian secara fungsional .



Gambar 3.20 Pemasangan Komponen