



**PERANCANGAN SISTEM KONTROL LAMPU BERBASIS
MIKROKONTROLER NODEMCU DENGAN
SENSOR SUARA**

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

SKRIPSI

OLEH

NAMA : RUDIYANTO
NPM : 1724370557
PROGRAM STUDI : SISTEM KOMPUTER

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

ABSTRAK

RUDIYANTO

Perancangan Sistem Kontrol Lampu Berbasis Mikrokontroler NodemCU dengan Sensor Suara 2019

Perancangan sensor suara untuk kontrol lampu menggunakan NodeMCU sebagai mikrokontroler. Ac light dimmer digunakan sebagai alat untuk menurunkan atau menaikkan intensitas cahaya sesuai dengan perintah suara yang dilakukan oleh pengguna dari jarak jauh. Perintah suara akan memudahkan, sehingga untuk menghidupkan lampu tidak harus berjalan menuju tempat saklar lampu. Perancangan sensor suara pada lampu dengan menggunakan NodemCU dan Ac light dimmer sangat efisien sehingga dapat digunakan bagi masyarakat luas. Kata kunci : *NodemCU, Ac light dimmer, sensor suara, saklar*

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR.....	v
DAFTAR TABEL	vi
DAFTAR LAMPIRAN	vii
DAFTAR ISTILAH	viii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
BAB II LANDASAN TEORI	
2.1 Sistem Kendali	4
2.2 Sistem Kontrol	4
2.3 Internet Of Things	5
2.4 ESP 8266 NoDemCU	6
2.5 A.T Command	10
2.6 App Inventor	11
2.7 Arduino IDE	12
2.8 Thingspeak(IOT)	20
2.9 Sensor Suara (<i>Speech Recognition</i>)	21
2.9.1 Skema utama Dan Algoritma Sensor Suara	22
2.10 AC Light Dimmer Module	23
2.11 Smartphone	24
2.11.1 Android	24
2.11.3 Sejarah dan Versi Android	25
2.11.4 Fitur Android	26
BAB III METODE PENELITIAN	
3.1 Jenis dan Metodologi Penelitian	28
3.2 Analisa Kebutuhan Sistem	31
3.2.1 Kebutuhan Fungsional Sistem	32
3.2.2 Kebutuhan Nonfungsional Sistem	32
3.3 Kebutuhan Perangkat Lunak	32
3.3.1 Perancangan Pemograman Perangkat Lunak Pada Sistem Tertanam	33
3.4 Perancangan User Interface	35
3.5 Perancangan Alur Kerja Web Server	39
3.5.1 Kebutuhan Perangkat Keras	40
3.6 Analisa Kebutuhan	41
3.6.1 Analisa Kebutuhan Penelitian	41
3.6.2 Analisa Kebutuhan Perangkat Keras	41

3.6.3 Analisa Kebutuhan Perangkat Lunak	41
---	----

BAB IV HASIL DAN PEMBAHASAN

4.1 Pengujian Zero Crossing Ditektor	42
4.2 Pengujian Dimer Lampu	43
4.3 Pengujian Perangkat Keras	45
4.4 Pengujian Perangkat Lunak	47
4.5 Pengiriman Nilai Data ke ThingSpeak	55
4.6 Pembacaan Data ThingSpeak Pada Modul NoDemCU	56
4.7 Pembuatan Aplikasi Android	57
4.8 Pengujian Secara Keseluruhan	60

BAB V PENUTUP

5.1 Kesimpulan	62
5.2 Saran	63

DAFTAR PUSTAKA

BIOGRAFI PENULIS

LAMPIRAN-LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi saat ini mendorong manusia untuk terus berpikir kreatif, tidak hanya menggali penemuan-penemuan baru, tapi juga memaksimalkan kinerja teknologi yang ada untuk meringankan kerja manusia dalam kehidupan sehari-hari seperti pengendalian lampu atau perangkat elektronik lainnya. (Setiawan, 2016)

Oleh karena itu apabila seluruh lampu pada suatu tempat baik itu di rumah, cafe, kantor dan lain sebagainya dikendalikan tanpa harus menyalakan saklar pada tempat tersebut maka peran mikrokontroler, *smartphone android*, serta fasilitas wifi sangat penting untuk memberi kenyamanan dan kemudahan khususnya, untuk penyandang cacat fisik atau orang yang sudah tua.

Berdasarkan latar belakang yang sudah dijelaskan maka penulis mengangkat judul tugas akhir yaitu **“PERANCANGAN SISTEM KONTROL LAMPU BERBASIS MIKROKONTROLER NodeMCU DENGAN SENSOR SUARA”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan masalah perancangan tugas akhir ini yaitu:

1. Bagaimana merancang sistem dan program kontrol lampu berbasis mikrokontroler NodeMCU.
2. Bagaimana mengaplikasikan sistem kontrol lampu pada android.

1.3 Batasan Masalah

Dengan permasalahan yang ada, maka dalam tugas akhir dibatasi mengenai :

1. Menggunakan *smartphone android* sebagai alat pengontrol lampu. Aplikasi *smartphone* didesain berdasarkan alat yang akan dibuat dengan bantuan *voice command* yang disediakan oleh *google voice*.
2. Menggunakan ESP8266 NodeMCU sebagai media transmisi antara *smartphone android* dengan ac light dimmer yang nantinya ESP8266 NodeMCU sebagai media *client* yang menghubungkan android dengan *webservice*.
3. Perancangan alat menggunakan NodeMCU sebagai mikrokontroler.

1.4 Tujuan Penelitian

Adapun tujuan dari pembuatan alat ini adalah:

1. Merancang system control lampu yang dapat digunakan dimana saja.
2. Memudahkan user dalam mengontrol lampu hanya dengan perintah suara.
3. Merancang system pengontrolan lampu agar dapat dikontrol dari jarak jauh menggunakan modul ESP8266.
4. Merancang bentuk fisik dari pembuatan *prototype* control lampu menggunakan sensor suara..
5. Merancang aplikasi pada *smartphone android* sebagai media control lampu yang dapat dipakai oleh semua user.

1.5 Manfaat Penelitian

Manfaat yang di dapat dalam pembuatan tugas akhir ini adalah :

1. Untuk mengefisienkan tenaga bagi pengguna dengan menggunakan alat ini.
2. Mengaplikasikan penggunaan system pengontrolan lampu yang berbasis *android* dikehidupan sehari-hari.
3. Merancang dan merangkai control lampu berbasis mikrokontroler NodeMCU dengan sensor suara.
4. Dapat mengontrol lampu rumah dimanapun user berada.
5. Menjadi telaah penelitian selanjutnya, khususnya pada perkembangan untuk peralatan listrik lainnya, dan juga dapat dikembangkan untuk gedung-gedung bertingkat.

BAB II

LANDASAN TEORI

2.1 Sistem Kendali

Sistem kendali dapat dikatakan sebagai hubungan antara komponen yang membentuk sebuah konfigurasi sistem yang akan menghasilkan tanggapan sistem yang diharapkan. Jadi harus ada yang dikendalikan, yang merupakan suatu sistem fisis, yang biasa disebut dengan kendalian (*plant*). Masukan dan keluaran merupakan variabel atau besaran fisis. Keluaran merupakan hal yang dihasilkan; sedangkan masukan adalah yang mempengaruhi kendalian, yang mengatur keluaran. Kedua dimensi masukan dan keluaran tidak harus sama.

Pada sistem kendali dikenal sistem lup terbuka (*open loop system*) dan sistem lup tertutup (*closed loop system*). Sistem kendali lup terbuka atau umpan maju (*feed forward control*) umumnya mempergunakan pengatur (*controller*) serta aktuator kendali (*control actuator*) yang berguna untuk memperoleh respon sistem yang baik. Sistem kendali ini keluarannya tidak diperhitungkan ulang oleh controller. Suatu keadaan apakah plant benar – benar telah mencapai target seperti yang dikehendaki masukan atau referensi tidak dapat mempengaruhi kinerja kontroler. (haris abdul, 2014)

2.2 Sistem Kontrol

Sistem kontrol adalah proses pengaturan ataupun pengendalian terhadap satu atau beberapa besaran (*variabel, parameter*) sehingga berada pada suatu harga atau dalam suatu rangkuman harga (*range*) tertentu. Didalam dunia industri,

dituntut suatu proses kerja yang aman dan berefisiensi tinggi untuk menghasilkan produk dengan kualitas dan kuantitas yang baik serta dengan waktu yang telah ditentukan. Otomatisasi sangat membantu dalam hal kelancaran operasional, keamanan (investasi, lingkungan), (biaya produksi), mutu produk. (haris abdul, 2014)

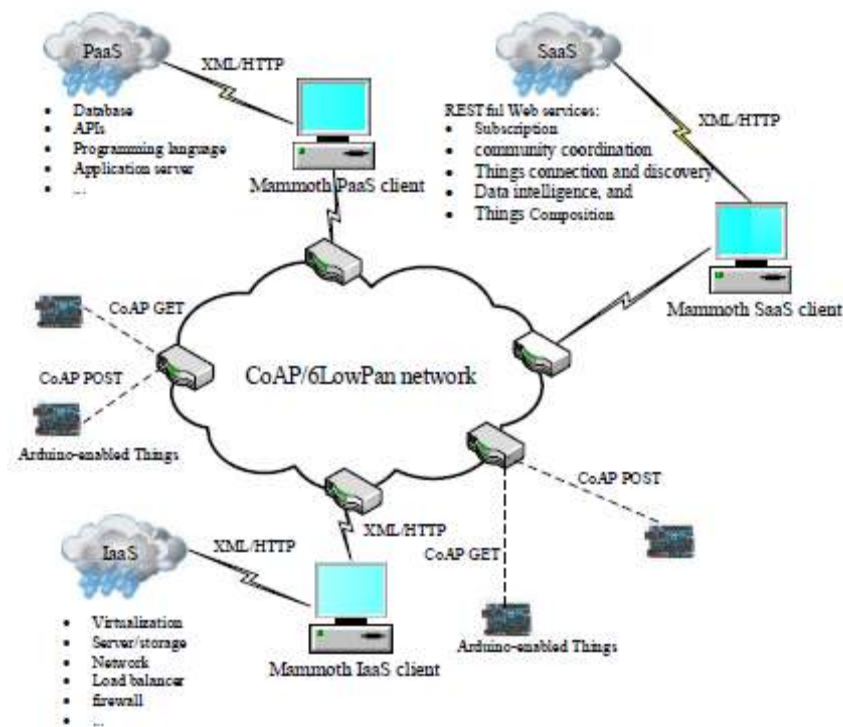
2.3 Internet of Things

IoT (*Internet of Things*) memungkinkan pengguna untuk mengelola dan mengoptimalkan elektronik dan peralatan listrik yang menggunakan internet. Hal ini berspekulasi bahwa sebagian waktu dekat komunikasi antara computer dan peralatan elektronik mampu bertukar informasi diantara mereka sehingga mengurangi interaksi manusia. Hal ini juga akan membuat pengguna internet semakin meningkat dengan berbagai fasilitas dan layanan internet. (Apri, 2015)

Tantangan utama dalam IOT adalah menjembatani kesenjangan antara dunia fisik dan dunia informasi. Seperti bagaimana mengolah data yang diperoleh dari peralatan elektronik melalui sebuah interface antara pengguna dan peralatan itu. Sensor mengumpulkan data mentah fisik dari skenario *real time* dan mengkonversikan ke dalam mesin format yang dimengerti sehingga akan mudah dipertukarkan antara berbagai bentuk format data (Thing) .

Perkembangan pada teknologi *mobile* juga ikut memberi sumbangsih kepada perkembangan *Internet of Things* yaitu dilakukannya penelitian tentang privasi di bidang pengamatan wilayah, mendeteksi lokasi berdasarkan *Location Based Service* sehingga seseorang bisa merasa nyaman menggunakan perangkat *mobile* tanpa harus terganggu privasi pribadi .

Isu *Cloud Computing* juga menjadi bahan penelitian *Internet of Things* dengan menggabungkan teknologi *cloud computing* dan *Internet of Things* yang disebut dengan *CloudThings*.



Gambar 2.1 Arsitektur CloudThings
(Sumber : Junaidi, 2015)

2.4 ESP8266 NodeMCU

ESP8266 varian ESP-12 yaitu NodeMCU yang merupakan platform *Internet of Things* yang bersifat *opensource*. Terdapat *hardware* berupa SoC (*System on Chip*) ESP8266, sehingga dapat melakukan programming langsung ke ESP8266 tanpa menggunakan mikrokontroler tambahan. Memprogram NodeMCU hanya diperlukan ekstensi kabel data USB yang sama persis dengan kabel *charging smartphone*, karena NodeMCU telah memnyusun ESP8266 ke dalam sebuah board dengan berbagai fitur layaknya mikrokontroler, serta akses terhadap WiFi

dengan *chip* komunikasi USB to serial. (Embeddednesia, 2017)



Gambar 2.2 ESP8266 NodeMCU

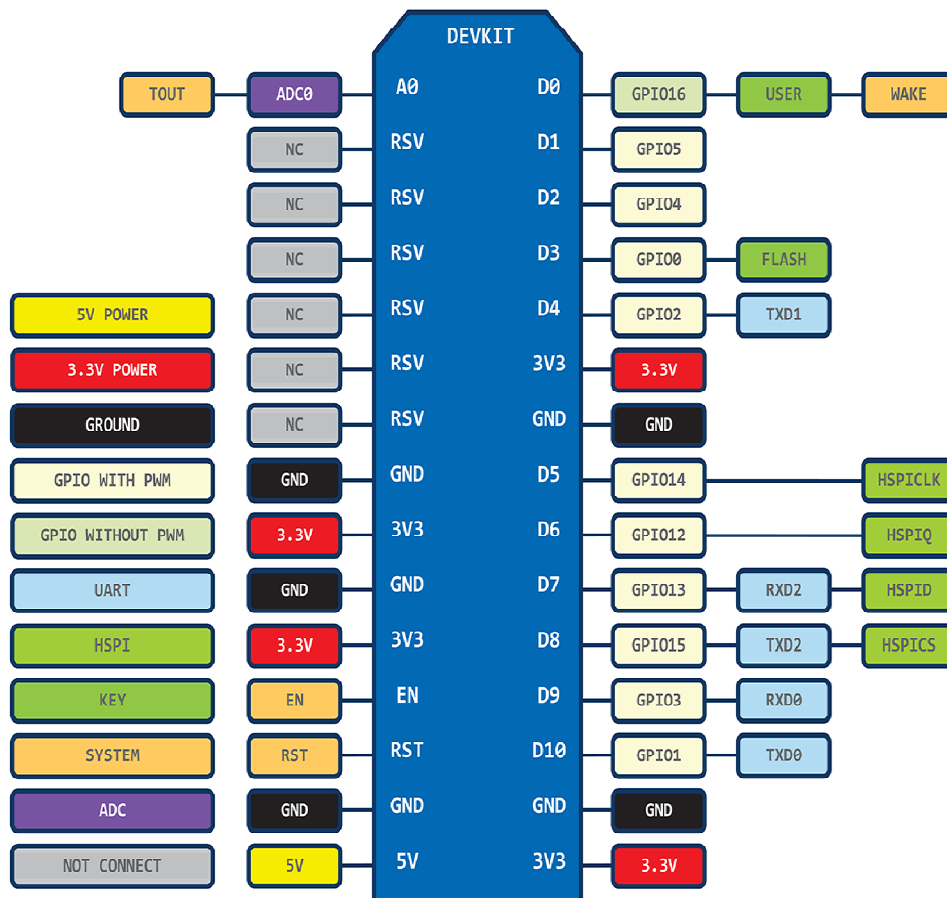
(Sumber : <https://embeddednesia.com/v1/tutorial-nodemcu-pertemuan-pertama/> Diakses 23-maret-2019)

NodeMCU berukuran panjang 4.83cm, lebar 2.54cm, dan berat 7 gram. Board ini sudah dilengkapi dengan fitur WiFi dan Firmwarena yang bersifat *opensource*.

Spesifikasi yang dimiliki oleh NodeMCU sebagai berikut :

- a. Board ini berbasis ESP8266 serial WiFi SoC (Single on Chip) dengan onboard USB to TTL. Wireless yang digunakan adalah IEEE 802.11b/g/n.
- b. 2 tantalum capacitor 100 micro farad dan 10 micro farad.
- c. 3.3v LDO regulator.
- d. Blue led sebagai indikator.
- e. Cp2102 usb to UART bridge.
- f. Tombol reset, port usb, dan tombol flash.
- g. Terdapat 9 GPIO yang di dalamnya ada 3 pin PWM, 1 x ADC Channel, dan pin RX TX
- h. 3 pin ground.
- i. S3 dan S2 sebagai pin GPIO

- j. S1 MOSI (Master Output Slave Input) yaitu jalur data dari master dan masuk ke dalam slave, sc cmd/sc.
- k. S0 MISO (Master Input Slave Input) yaitu jalur data keluar dari slave dan masuk ke dalam master.
- l. SK yang merupakan SCLK dari master ke slave yang berfungsi sebagai clock.
- m. Pin Vin sebagai masukan tegangan.
- n. Built in 32-bit MCU.



Gambar 2.3 GPIO NodeMCU ESP8266 v2

(Sumber : <https://embeddednesia.com/v1/tutorial-nodemcu-pertemuan-pertama/> Diakses 23-maret-2019)

- RST : berfungsi mereset modul
- ADC: Analog Digital Converter. Rentang tegangan masukan 0-1v, dengan skup nilai digital 0-1024
- EN: Chip Enable, Active High
- IO16 :GPIO16, dapat digunakan untuk membangunkan chipset dari mode deep sleep
- IO14 : GPIO14; HSPI_CLK

- f. IO12 : GPIO12: HSPI_MISO
- g. IO13: GPIO13; HSPI_MOSI; UART0_CTS
- h. VCC: Catu daya 3.3V (VDD)
- i. CS0 :Chip selection
- j. MISO : Slave output, Main input
- k. IO9 : GPIO9
- l. IO10 GBIO10
- m. MOSI: Main output slave input
- n. SCLK: Clock
- o. GND: Ground
- p. IO15: GPIO15; MTDO; HSPICS; UART0_RTS
- q. IO2 : GPIO2;UART1_TXD
- r. IO0 : GPIO0
- s. IO4 : GPIO4
- t. IO5 : GPIO5
- u. RXD : UART0_RXD; GPIO3
- v. TXD : UART0_TXD; GPIO1

2.5 AT Command

Modul *wireless* ESP8266 yang sudah tersedia pada *NodeMCU board* memiliki *firmware* bawaan pabrik yang mendukung perintah AT-Command. Sekumpulan daftar hari Hayes command merupakan deskripsi dari AT-Command. Hayes command dikembangkan oleh Dennis Hayes pada tahun 1981

sebagai daftar perintah untuk melakukan konfigurasi modem dengan menggunakan jalur serial interface.(Yuliansyah, 2016)

Berikut ini beberapa contoh *Hayes Command* beserta fungsinya pada modul ESP8266.

Tabel 2.1 Daftar AT-Command

AT-Command	Function	Response
AT	Working	OK
AT+RST	Restart	OK [System Ready, Vendor:www.ai-thingker.com]
AT+GMR	Firmware Version	AT+GMR 0018000902 OK
AT+CWLAP	List Access	AT+CWLAP+CWLAP:(4,"AP 1",-38,"70:62:b8:6f:6d58",1)+CWLAP:(4,"AP 2",-83,"f8:7b:8c:1e:7c:6d",1)
AT+CWJAP? AT+CWJAP="SSID","PASS"	Join Access Point	Query AT+CWJAP?+CWJAP:"AP 1" OK

Sumber :

(https://www.sparkfun.com/datasheets/.../AT_Commands_Reference_Guide_r0.pdf/diakses 20-mei-2019)

2.6 App inventor

App Inventor adalah sebuah pemrograman visual yang digunakan untuk mengembangkan aplikasi berbasis android dengan dukungan fitur berupa drag - drop tool. App Inventor for android adalah aplikasi yang pada dasarnya disediakan oleh Google dan sekarang di-Maintenance oleh *Massachusetts Institute of Technology* (MIT) Komputer (appinventor, 2019). App Inventor memungkinkan semua orang untuk membuat *software* aplikasi untuk sistem operasi android. Pengguna dapat menggunakan tampilan grafis GUI dan fitur

drag drop visual objek untuk membuat sebuah aplikasi dapat berjalan pada sistem operasi android (Fitria:2016).



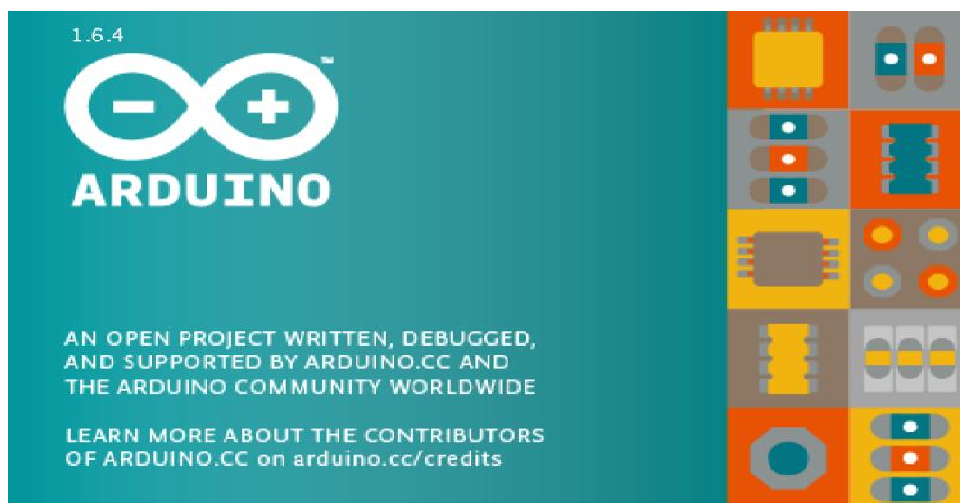
Gambar 2.4 : Tampilan App Inventor
sumber: (ai2.appinventor.mit.edu)

App Inventor juga sebuah aplikasi *builder* untuk membuat aplikasi yang berjalan di sistem operasi Android yang disediakan oleh googlelabs. Jadi kalian harus punya *account* google dulu untuk bisa masuk ke *home app inventor*. App Inventor ini sedikit berbeda dengan *app builder* lain. Dengan App Inventor kita tidak pernah menemui kasus para developer uring-uringan gara-gara aplikasi yang dibuat nggak jalan, dan ternyata itu hanya karena kesalahan sintak kurang tanda semicolon (;). App Inventor ini menggunakan teknik visual programming, berbentuk seperti susunan *puzzle-puzzle* yang memiliki logika tertentu. Kalian bisa baca-baca tutorial dan contoh-contoh aplikasi nya di *page Explore*.

2.7 Arduino IDE

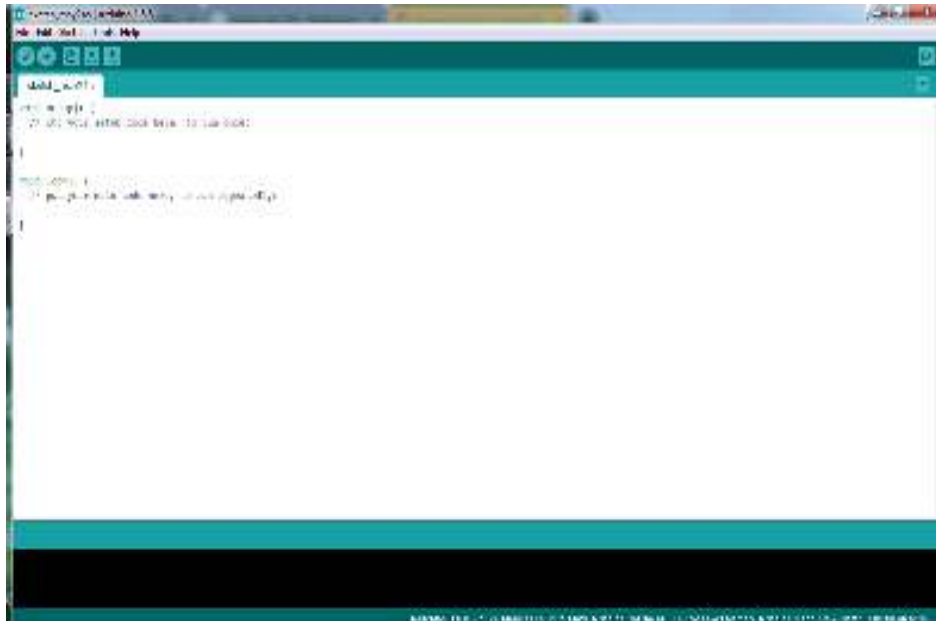
Arduino IDE arduino adalah *software* yang sangat canggih ditulis dengan menggunakan java. IDE Arduino terdiri dari : - Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa

Processing. - 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 *microcontroller* adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini. - Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memory dalam papan arduino.(Baidhowi & Neforawati, 2017)



Gambar 2.5 : Arduino IDE

Sumber : (Baidhowi : 2017)



Gambar 2.6 : Software Arduino IDE

(Sumber : <https://www.sinuarduino.com/artikel/mengenal-arduino-software-ide/>.

Diakses 23-maret-2019)

Gambar diatas merupakan tampilan dari Software Arduino IDE :



Verify

Berfungsi untuk melakukan checking kode yang kamu buat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum.



Upload

Berfungsi untuk melakukan kompilasi program atau kode yang kamu buat menjadi bahasa yang dapat dipahami oleh mesih alias si Arduino.



New

Berfungsi untuk membuat *Sketch* baru.



Open

Berfungsi untuk membuka *sketch* yang pernah kamu buat dan membuka kembali untuk dilakukan editing atau sekedar upload ulang ke Arduino.



Save

Berfungsi untuk menyimpan *Sketch* yang telah kamu buat.



Serial Monitor

Berfungsi untuk membuka serial monitor. Serial monitor disini merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan sketch pada port serialnya. Serial Monitor ini sangat berguna sekali ketika kamu ingin membuat program atau melakukan *debugging* tanpa menggunakan LCD pada Arduino. Serial monitor ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan error.

a. File

- *New*, berfungsi untuk membuat membuat sketch baru dengan bare minimum yang terdiri void setup() dan void loop().
- *Open*, berfungsi membuka sketch yang pernah dibuat di dalam drive.
- *Open Recent*, merupakan menu yang berfungsi mempersingkat waktu pembukaan file atau sketch yang baru-baru ini sudah dibuat.
- *Sketchbook*, berfungsi menunjukkan hirarki *sketch* yang kamu buat termasuk struktur foldernya.

- **Example**, berisi contoh-contoh pemrograman yang disediakan pengembang Arduino, sehingga kamu dapat mempelajari program-program dari contoh yang diberikan.
- **Close**, berfungsi menutup jendela Arduino IDE dan menghentikan aplikasi.
- **Save**, berfungsi menyimpan *sketch* yang dibuat atau perubahan yang dilakukan pada *sketch*
- **Save as**, berfungsi menyimpan *sketch* yang sedang dikerjakan atau *sketch* yang sudah disimpan dengan nama yang berbeda.
- **Page Setup**, berfungsi mengatur tampilan page pada proses pencetakan.
- **Print**, berfungsi mengirimkan file *sketch* ke mesin cetak untuk dicetak.
- **Preferences**, disini kamu dapat merubah tampilan *interface* IDE Arduino.
- **Quit**, berfungsi menutup semua jendela Arduino IDE. *Sketch* yang masih terbuka pada saat tombol *Quit* ditekan, secara otomatis akan terbuka pada saat Arduino IDE dijalankan.

b. Edit

- **Undo/Redo**, berfungsi untuk mengembalikan perubahan yang sudah dilakukan pada *Sketch* beberapa langkah mundur dengan *Undo* atau maju dengan *Redo*.
- **Cut**, berfungsi untuk *remove* teks yang terpilih pada editor dan menempatkan teks tersebut pada *clipboard*.

- **Copy**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard*.
- **Copy for Forum**, berfungsi melakukan *copy* kode dari editor dan melakukan *formatting* agar sesuai untuk ditampilkan dalam forum, sehingga kode tersebut bisa digunakan sebagai bahan diskusi dalam forum.
- **Copy as HTML**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard* dalam bentuk atau format HTML. Biasanya ini digunakan agar code dapat *diembedddkan* pada halaman web.
- **Paste**, berfungsi menyalin data yang terdapat pada *clipboard*, kedalam editor.
- **Select All**, berfungsi untk melakukan pemilihan teks atau kode dalam halaman editor.
- **Comment/Uncomment**, berfungsi memberikan atau menghilangkan tanda *//* pada kode atau teks, dimana tanda tersebut menjadikan suatu baris kode sebagai komen dan tidak disertakan pada tahap kompilasi.
- **Increase/Decrease Indent**, berfungsi untuk mengurangi atau menambahkan indentntasi pada baris kode tertentu. Indentasi adalah “tab”.
- **Find**, berfungsi memanggil jendela *window find and replace*, dimana kamu dapat menggunakannya untuk menemukan variabel atau kata tertentu dalam program atau menemukan serta menggantikan kata tersebut dengan kata lain.

- ***Find Next***, berfungsi menemukan kata setelahnya dari kata pertama yang berhasil ditemukan.
- ***Find Previous***, berfungsi menemukan kata sebelumnya dari kata pertama yang berhasil ditemukan

c. Sketch

- ***Verify/Compile***, berfungsi untuk mengecek apakah *sketch* yang kamu buat ada kekeliruan dari segi sintaks atau tidak. Jika tidak ada kesalahan, maka sintaks yang kamu buat akan dikompilasi ke dalam bahasa mesin.
- ***Upload***, berfungsi mengirimkan program yang sudah dikompilasi ke Arduino Board.
- ***Upload Using Programmer***, menu ini berfungsi untuk menuliskan *bootloader* ke dalam IC Mikrokontroler Arduino. Pada kasus ini kamu membutuhkan perangkat tambahan seperti *USBasp* untuk menjembatani penulisan program *bootloader* ke IC Mikrokontroler.
- ***Export Compiled Binary***, berfungsi untuk menyimpan file dengan ekstensi **.hex**, dimana file ini dapat disimpan sebagai arsip untuk di upload ke board lain menggunakan tools yang berbeda.
- ***Show Sketch Folder***, berfungsi membuka folder *sketch* yang saat ini dikerjakan.
- ***Include Library***, berfungsi menambahkan library/pustaka ke dalam *sketch* yang dibuat dengan menyertakan sintaks `#include` di awal

kode. Selain itu kamu juga bisa menambahkan library eksternal dari file **.zip** kedalam Arduino IDE.

- **Add File...**, berfungsi untuk menambahkan file kedalam *sketch* arduino (file akan dikopikan dari drive asal). File akan muncul sebagai tab baru dalam jendela *sketch*.

d. Tools

- **Auto Format**, berfungsi melakukan pengatran format kode pada jendela editor
- **Archive Sketch**, berfungsi menyimpan sketch kedalam file **.zip**
- **Fix Encoding & Reload**, berfungsi memperbaiki kemungkinan perbedaan antara pengkodean peta karakter editor dan peta karakter sistem operasi yang lain.
- **Serial Monitor**, berfungsi membuka jendela serial monitor untuk melihat pertukaran data.
- **Board**, berfungsi memilih dan melakukan konfigurasi board yang digunakan.
- **Port**, memilih port sebagai kanal komunikasi antara software dengan hardware.
- **Programmer**, menu ini digunakan ketika kamu hendak melakukan pemrograman chip mikrokontroler tanpa menggunakan koneksi Onboard USB-Serial. Biasanya digunakan pada proses *burning bootloader*.

- **Burn Bootloader**, mengizinkan kamu untuk mengkopikan program bootloader kedalam IC mikrokontroler

e. **Help**

Disini kamu bisa mendapatkan bantuan terhadap kegalauanmu mengenai pemrograman. Menu help berisikan file-file dokumentasi yang berkaitan dengan masalah yang sering muncul, serta penyelesaiannya. Selain itu pada menu help juga diberikan link untuk menuju Arduino Forum guna menanyakan serta mendiskusikan berbagai masalah yang ditemukan.(Sinuarduino, 2016)

2.8 **ThingSpeak (IOT)**

Thingspeak merupakan web berbasis open API IOT source platform informasi yang komprehensif dalam menyimpan data sensor dari 'aplikasi IOT' bervariasi dan berkonspirasi, data *output* yang dihasilkan dalam bentuk grafik di tingkat web. Thingspeak berkomunikasi dengan bantuan koneksi internet yang bertindak sebagai „paket data“ pembawa antara „hal/benda (sensor)“ yang saling terhubung dan Thingspeak mengambil, menyimpan, menganalisis, mengamati dan bekerja pada data yang dirasakan dari sensor yang terhubung ke mikrokontroler seperti 'Arduino, modul TI CC3200, Raspberry-pi, esp8266.(Samsugi & Kastutara, 2018)



Gambar 2.7 : ThingSpeak

Sumber : (<https://www.nyebarilmu.com/project-nodemcu-membaca-sensor-dht11-dengan-thingspeak/> :2019, diakses 20-mei-2019)

2.9 Sensor Suara (Speech Recognition)

Sensor Suara adalah proses identifikasi suara berdasarkan kata yang diucapkan dengan melakukan konversi sebuah sinyal akustik, yang ditangkap oleh *audio device* (perangkat input suara). *Speech Recognition* juga merupakan sistem yang digunakan untuk mengenali perintah kata dari suara manusia dan kemudian diterjemahkan menjadi suatu data yang dimengerti oleh komputer. Pada saat ini, sistem ini digunakan untuk menggantikan peranan input dari *keyboard* dan *mouse*.

Keuntungan dari sistem ini adalah pada kecepatan dan kemudahan dalam

penggunaannya. Kata – kata yang ditangkap dan dikenali bisa jadi sebagai hasil akhir, untuk sebuah aplikasi seperti *command & control*, penginputan data, dan persiapan dokumen. Parameter yang dibandingkan ialah tingkat penekanan suara yang kemudian akan dicocokkan dengan template database yang tersedia. Sedangkan sistem pengenalan suara berdasarkan orang yang berbicara dinamakan *speaker recognition*. Pada tulisan ini hanya akan dibahas mengenai *speech recognition* karena kompleksitas algoritma yang diimplementasikan lebih sederhana daripada *speaker recognition*.(Rasimah, 2014)

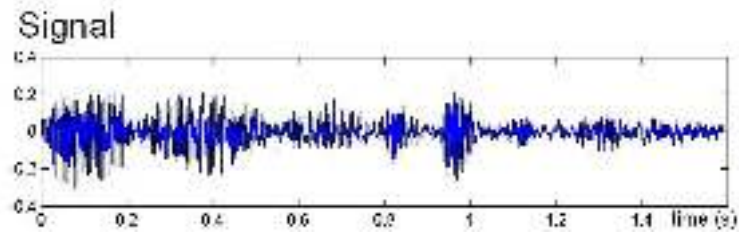
2.7.1 Skema utama dan algoritma Sensor Suara

Terdapat empat langkah utama dalam sistem pengenalan suara yaitu :

1. Penerima data input
2. Ekstraksi, yaitu penyimpanan data masukan sekaligus pembuatan database untuk template.
3. Perbandingan/pencocokan, yaitu tahap pencocokan data baru dengan data suara (pencocokan tata bahasa) pada template.
4. Validasi identitas pengguna.

Tahap ini merupakan tahapan pencocokan data baru dengan data yang data suara (Pencocokan tata bahasa) pada pola.

Secara umum *speech recognition* memproses sinyal suara yang masuk dan menyimpan dalam bentuk digital. Hasil proses digitalisasi tersebut kemudian dikonversi dalam bentuk spektrum suara yang akan dianalisa dengan membandingkannya dengan templet suara pada database sistem.



Gambar 2.8 :Sinyal suara
Sumber : (Rasimah, 2014)

2.10 AC Light Dimmer Module

AC Light Dimmer modul dimmer buatan RobotDyn yang dapat dikontrol oleh mikrokontroler seperti Arduino, Raspberry Pi dan sebagainya. Dengan adanya fitur pin *zero crossing detector* di modul ini, membuat mikrokontroler dapat mengetahui saat yang tepat untuk mengirim sinyal PWM. (Tanpa timing yang tepat, arus AC dengan TRIAC akan menghasilkan sinyal output yang kacau bila dihubungkan dengan PWM dan dapat menyebabkan dimmer tidak berfungsi dalam menghasilkan sinyal PSM dengan kurva yang benar.(kurniawan adris, 2014) Gambar AC light dimmer module dapat di lihat di gambar 2.9 dibawah ini :



Gambar 2.9 : AC Light Dimmer Module
Sumber : (Kurniawan:2018)

Dimmer juga merupakan rangkaian elektronik yang memodifikasi bentuk sinyal ac murni menjadi sinyal terpotong-potong sehingga daya keluaran bisa

diatur. Pemotongan sinyal ac ini berguna sebagai peredup lampu, memperlambat motor, mengatur pemanasan dan lainnya. Dimmer yang lebih kompleks menggunakan PWM sebagai pengendalinya. PWM bisa dihasilkan oleh rangkaian SCR, chip/IC PWM atau mikrokontroler. Dimmer PWM ini mampu menghasilkan tingkatan daya yang kecil, sehingga pengontrolan menjadi lebih presisi.

Dimmer PWM bisa dikategorikan menjadi dua macam yaitu :

1. Penyalaan berdasarkan titik nol.
2. Penyalaan bebas.

2.11 Smartphone

Smartphone merupakan kombinasi fungsi dari perangkat komunikasi dan perangkat penunjang kebutuhan digital *lifestyle* dengan beberapa fitur multimedia dan organizer. Seiring perkembangan jaman, *smartphone* sekarang ditunjang dengan fitur GPS untuk navigasi, NFC untuk komunikasi instan dalam pertukaran data. Pada umumnya *smartphone* memiliki prosesor yang cukup tinggi berkat teknologi *SoC (System on Chip)* yang menghadirkan kemampuan *hardware* yang tinggi namun dengan ukuran yang kompak.(Dimas, Sadewo, Widasari, & Muttaqin, 2017)

2.11.1 Android

Android merupakan salah satu operasi system pada perangkat *mobile*. Dalam pengembangan aplikasi *android* menggunakan platform java sebagai bahasa pemrogramannya. Google bekerjasama dengan lebih dari 47

perusahaan lain yang tergabung dalam OHA yaitu (*Open Handset Alliance*) untuk membuat standar pada perangkat *mobile*.



Gambar 2.10 : Simbol Android
Sumber : (Sadewo : 2017)

2.11.2 Sejarah dan versi *android*

Android sebagai sistem operasi pertama kali ada pada tahun 2003 yang dikembangkan oleh perusahaan Android inc. Pada tahun 2006 perusahaan raksasa google mengambil alih perusahaan tersebut.

Pada tanggal 12 November 2007 pertama kali dirilis *SDKAndroidBeta*, perangkat *mobile* yang pertama kali menggunakan *android* adalah HTC dengan sistem operasi *android* 1.0 resmi dirilis pada tanggal 23 september 2008. Kemudian android berkembang dan mulai dipakai beberapa manufaktur *smartphone* dunia. Berikut sejarah perkembangan versi android pada tabel.

Tabel 2.2 : Sejarah Versi Android

Versi/Codename	Tgl. Rilis
1.0 (Alpha)	23 September 2008
1.1(Beta)	9 Februari 2009
1.5 (Cupcake)	27 April 2009
1.6 (Donut)	15 September 2009
2.0 – 2.1 (Éclair)	26 Oktober 2009
2.2 – 2.2.3 (Froyo)	20 Mei 2010

2.3 – 2.3.7 (Gingerbread)	6 Desember 2010
3.0 – 3.2.6 (Honeycomb)	22 Februari 2011
4.0 – 4.0.4 (Ice Cream Sandwich)	18 Oktober 2011
4.1 – 4.3.1 (Jelly Bean)	9 Juli 2012
4.4 – 4.4.4 (Kit – kat)	31 Oktober 2013
5.0 – 5.1.1 (Lollipop)	12 November 2014
6.0 – 6.0.1 (Marshmallow)	5 Oktober 2015
7.0 – 7.1.1 (Nougat)	22 Agustus 2016

Antarmuka pengguna android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan didunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. Android adalah sistem operasi dengan sumber terbuka (open source), dan google merilis kodenya dibawah lisensi apache.

2.11.3 Fitur Android

Berikut ini adalah beberapa fitur yang terdapat pada *smartphone* dengan operating sistem *android*:

1. Touch screen. Dengan menggunakan fitur ini, mengakibatkan navigasi menu menjadi lebih mudah dan efisien dengan hanya menyentuh layar.
2. Multipage. Sangat berguna untuk keperluan multitasking. Pengguna dapat berpindah page tanpa perlu menutup page sebelumnya untuk digunakan nanti.
3. Merupakan sistem operasi terbuka (open source) sehingga dapat dimodifikasi bahkan membuat dengan bebas.
4. Memiliki kualitas grafik dan suara yang bagus karena standar yang digunakan seperti MP3 dengan grafik 3D.

5. Terdapat perangkat pendukung seperti, wifi, bluetooth, kamera dan GPS

BAB III

METODE PENELITIAN

3.1 Jenis dan Metodologi Penelitian

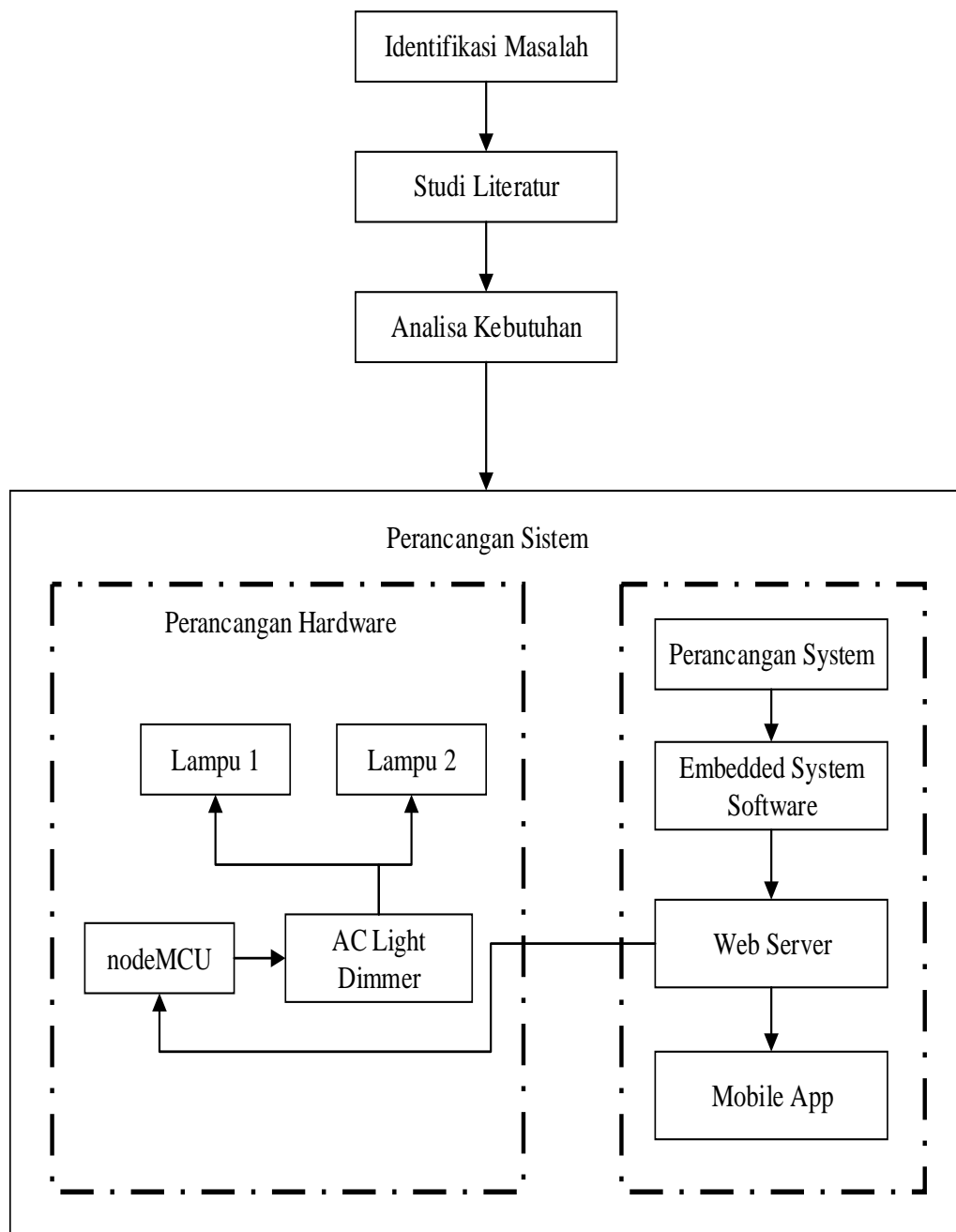
Jenis penelitian yang digunakan dalam penulisan tugas akhir ini adalah penelitian eksperimental (*Experimental Research*). Penelitian eksperimental adalah jenis penelitian yang digunakan untuk melihat hubungan sebab akibat. Penelitian eksperimental merupakan kegiatan penelitian yang bertujuan untuk menilai pengaruh suatu perlakuan atau tindakan dibandingkan dengan tindakan lain.

Penelitian eksperimental menggunakan sesuatu percobaan yang dirancang secara khusus guna membangkitkan data yang diperlukan untuk menjawab pertanyaan penelitian. Penelitian eksperimental dilakukan secara sistematis, logis, dan teliti di dalam melakukan kontrol terhadap kondisi.

Pada penelitian ini dilakukan penghubungan komponen alat-alat yang berbeda karakteristik. Penelitian ini bertujuan untuk mempelajari sesuatu dengan memvariasikan beberapa kondisi dan mengamati efek yang terjadi.

Penelitian ini ditunjang dengan studi literatur (*literatur research*), yaitu dengan membaca dan mempelajari literatur tentang *Perancangan Sistem Kontrol Lampu Berbasis Mikrokontroler NodeMCU Dengan Sensor Suara* serta berbagai komponen yang dibutuhkan dalam pengontrolan.

Metodologi penelitian dapat dilihat pada gambar 3.1:



Gambar 3.1. Metodologi Penelitian

Berdasarkan Gambar 3.1 dapat dijelaskan tahap-tahap yang akan dilakukan untuk menyelesaikan penelitian ini, yaitu:

a. Identifikasi Masalah

Identifikasi masalah dilakukan dengan cara pengumpulan materi berupa masalah melalui jurnal atau penelitian sebelumnya sehingga dengan melakukan pembuatan tugas akhir diharapkan dapat memberikan solusi untuk masalah tersebut. Dalam hal ini yaitu, adanya pengembangan dari penelitian sebelumnya yaitu *Smartroom Berbasis Android Dengan Teknologi Bluetooth Menggunakan Voice Command*.

b. Studi Literatur

Studi Literatur dilakukan dengan mencari serta mengumpulkan teori-teori yang mendukung dan berkaitan dengan pembuatan tugas akhir. Teori-teori tersebut meliputi *AC Light Dimmer Module*, *NodemCU*, *ESP8266*, *Arduino IDE*, *Internet of Thing* dan *Web Server*, dan *Android*.

c. Analisis kebutuhan

Untuk memenuhi kebutuhan sistem ini, maka sistem yang dirancang memenuhi dua fungsionalitas sistem yaitu kontrol lampu bisa memberikan notifikasi kepada *user* melalui *smartphone*.

d. Perancangan Sistem

Perancangan Sistem dibagi menjadi 2 tahap, yaitu:

a) Perancangan *Hardware*

Sistem membutuhkan sebuah ruangan lingkungan sistem (*plant*). Untuk menerapkan sistem *controller*, dibutuhkan perangkat keras yang terdiri

dari *AC Light Dimmer Module*, *ESP 8266 NodemCU* dan perangkat *smartphone*.

b) Perancangan *Software*

Perancangan *Software* meliputi proses pembacaan pengolahan data di mikrokontroller, pengiriman data ke *web server*, pengiriman data ke *smartphone* dan pengiriman notifikasi dan rekomendasi ke *smartphone*.

e. Implementasi Sistem

Tahapan implementasi Sistem menggambarkan proses implementasi perancangan penelitian yaitu, sistem control lampu dengan perintah suara pada di rumah.

f. Pengujian Sistem

Serangkaian pengujian terhadap sistem dilakukan untuk menguji kinerja dari masing-masing komponen yang membangun sistem *control* lampu.

g. Analisa Hasil

Dari pengujian sistem, dilakukan analisis kinerja sistem dan data-data yang didapatkan selama pengujian.

h. Dokumentasi Tugas Akhir

Dokumentasi dilakukan sebagai pelaporan hasil penelitian tugas akhir.

3.2 Analisa Kebutuhan Sistem

Untuk mempermudah menganalisis sebuah sistem dibutuhkan dua jenis kebutuhan. Kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya

dilakukan oleh sistem. Sedangkan kebutuhan nonfungsional adalah kebutuhan yang menitik beratkan pada properti perilaku yang dimiliki oleh sistem.

3.2.1 Kebutuhan Fungsional Sistem

Pada bagian ini akan dijelaskan kebutuhan dari sistem berupa hal yang dapat dilakukan oleh *user* nantinya, kebutuhan fungsional sistem ini diantaranya :

- a) *User* dapat mengontrol lampu rupa dengan suara.
- b) *User* dapat menerima notifikasi di *user smartphone*.

3.2.2 Kebutuhan Nonfungsional Sistem

Kebutuhan Non Fungsional berupa batasan layanan atau fungsi yang ditawarkan sistem, kebutuhan non fungsional dari sistem ini diantaranya :

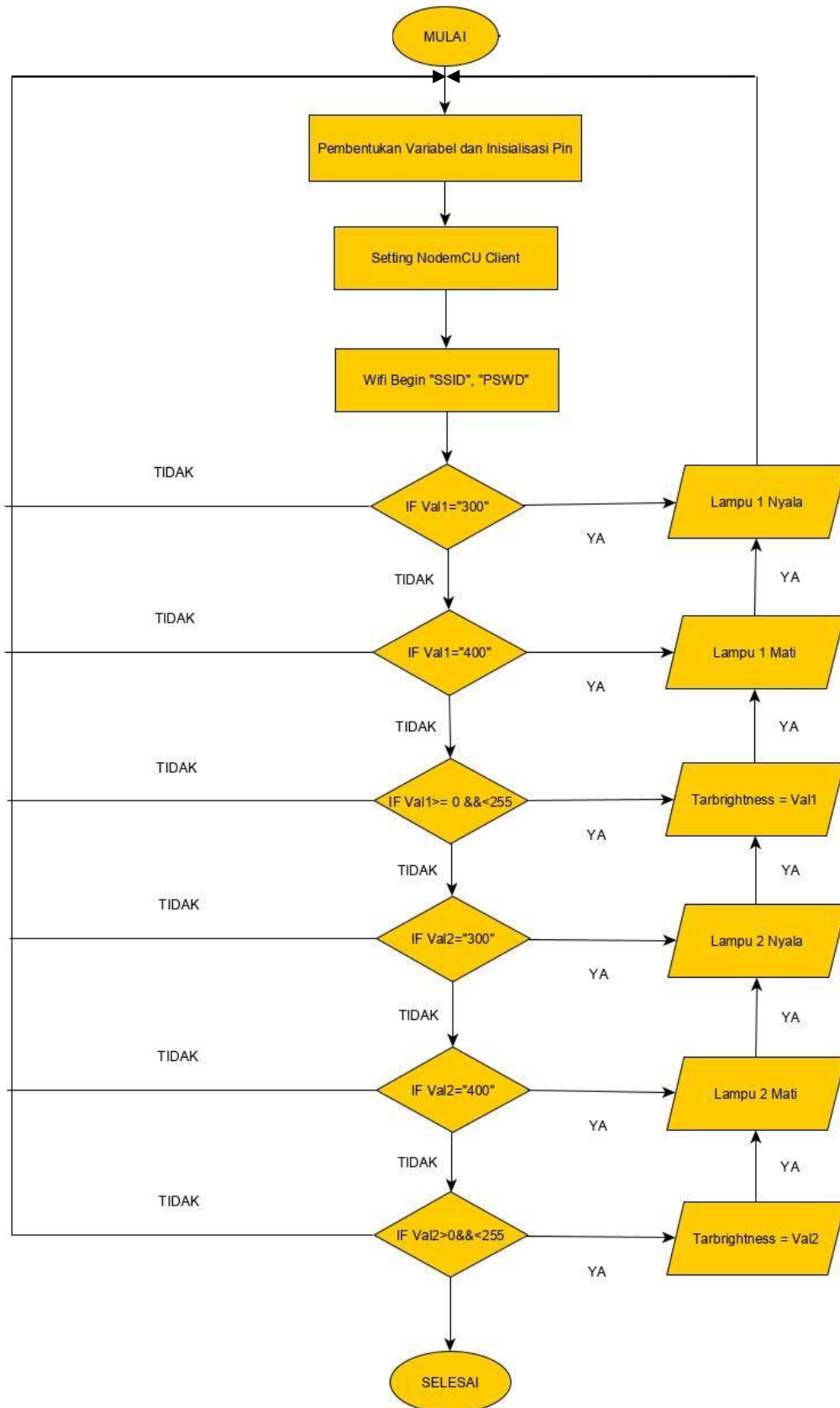
- a) Sistem dapat mengirim data secara *realtime*.
- b) Kegagalan sistem terjadi saat *web server* dalam keadaan *off*.
- c) Fungsi *monitoring* berjalan di *smartphone*.
- d) *Response time* dipengaruhi oleh koneksi ke *server*.

3.3 Kebutuhan Perangkat Lunak

Perancangan perangkat lunak dalam penelitian meliputi perancangan program yang menggunakan Arduino IDE sebagai *software* program yang berjalan di mikrokontroler. Pada pemograman perangkat lunak dibangun sebuah Aplikasi *interface* yang dapat menampilkan kontrol lampu pada *smart phone*

3.3.1 Perancangan Pemrograman Perangkat Lunak Pada Sistem Tertanam

Perancangan pemograman *embeddedsoftware* terdapat *flowchart* utama, Berikut Gambar 3.2 menggambarkan *flowchart* Perancangan pemograman Perangkat Lunak.



Gambar 3.2. Flowchart Utama Pemrograman Alat

Keterangan :

Pada saat memulai sistem kontrol, pada proses inisialisasi pin, Hal pertama yang dilakukan adalah setting NodemCu untuk mengatur Ssid dan Pswd agar NodemCu dengan Aplikasi android bisa tersambung, Kemudian setelah itu melakukan inisialisasi bahwa jika variabel val1 sama dengan 300 maka lampu nyala dan jika variabel val1 sama dengan 400 maka lampu mati dan jika variabel val1 lebih besar dari 0 dan lebih kecil dari 255 maka lampu terang atau samar.

3.4 Perancangan *User Interface*

Untuk aplikasi, sistem menggunakan aplikasi *android* yang dibuat menggunakan *APP Inventor* dengan berbasis pemograman java. Aplikasi dirancang untuk 1 *user* agar keamanan *rumah* tetap terjaga, aplikasi juga sederhana dan mudah dimengerti agar *user* dapat menggunakannya dengan mudah dan nyaman dalam menggunakan aplikasi ini sebagai *user interface* dari sistem. Tampilan Halaman Utama Aplikasi dapat dilihat pada Gambar 3.3-3.6.

a. Form Login



Gambar 3.3. Login Aplikasi

Form login berisi menu untuk *input username* dan *password*. Jika pengguna aplikasi belum mempunyai *username* dan *password*, maka pengguna diwajibkan untuk membuat akun baru terlebih dahulu. Pada tampilan form login terdapat *txtUsername* untuk memasukkan username, *txtPass* untuk memasukkan password, *btnLogin* untuk masuk ke sistem aplikasi, *checkbox1* sebagai pengingat akun, *btndaftar* untuk mendaftarkan akun baru, *btnVerTDB* sebagai tombol verifikasi akun dan *btnHapusTDB* sebagai tombol untuk menghapus akun yang sudah tidak digunakan.

b. Form Daftar

The image shows a mobile application interface for creating a new account. At the top, there is a logo featuring a red star with a yellow center, surrounded by a blue and green wreath. Below the logo, the text "Buat Akun Baru" is displayed. There are three input fields: "Nama Lengkap", "Username", and "Password". At the bottom, there are two buttons: "Simpan" (Save) and "Cancel". The background of the form is yellow.

Gambar 3.4. Form Daftar

Tampilan diatas adalah tampilan Form untuk mendaftarkan atau membuat akun baru. Pada tampilan form daftar terdapat *txtUsernamebaru* untuk memasukkan nama lengkap, *txtusrbaru* untuk memasukkan username, *txtpassbaru* untuk memasukkan password baru. Kemudian ada *btnsimpan* untuk menyimpan username dan password yang baru dibuat, dan *btncancel* untuk membatalkan membuat username baru.

c. Form Kontrol Sensor Suara



Gambar 3.5 Kontrol Sensor Suara

Setelah form login dan form untuk membuat username baru selesai, halaman berikutnya yang akan ditampilkan adalah form kontrol suara. Terdapat 3 *button* yang dapat dipilih yaitu *button* saklar suara sebagai sistem utama dari aplikasi, *button* credit yang menampilkan keterangan aplikasi, dan *button* kembali untuk kembali ke form login.

d. Form Credit



Gambar 3.6 Tentang Pembuat

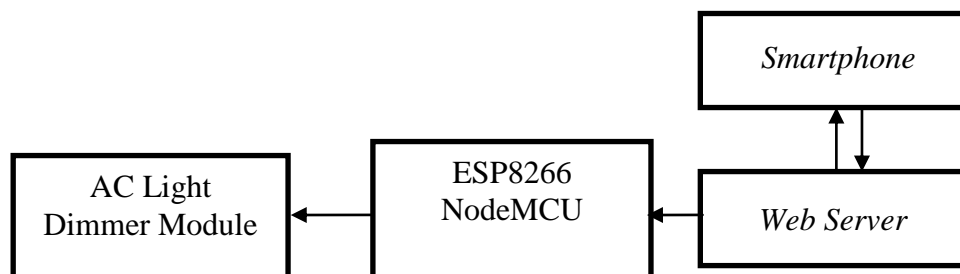
Form terakhir yang dirancang adalah form kredit. Form ini berisi informasi tentang aplikasi dan informasi tentang pembuat aplikasi.

3.5 Perancangan alur kerja *web server*

Web programming yang digunakan adalah PHP dan MySQL. PHP dan MySQL dibutuhkan untuk membuat *database* yang dapat menyimpan data berupa suhu dan gas berupa tanggal, bulan dan tahun, serta jam. Data yang tersimpan pada *database* tersebut dapat dilihat oleh petugas *User* dari aplikasi *mobile*.

3.5.1 Kebutuhan Perangkat Keras

Bentuk perancangan perangkat keras dari sistem *Perancangan Sistem Kontrol Lampu Berbasis Mikrokontroler NodemCU Dengan Sensor Suara*, dapat dilihat pada skema berikut ini:



Gambar 3.7 Skema Perancangan Komponen *Hardware* Sistem

Pada rancangan diatas, *smartphone* android akan dimanfaatkan sebagai penerjemah suara yang diinputkan dan akan merubahnya menjadi kode-kode digital. Dengan memanfaatkan salah satu *web server ThingSpeak.com* yang dapat digunakan sebagai *Internet of Things (IoT)*. Suara yang diinputkan nantinya akan diteruskan ke *nodeMCU* yang juga harus terhubung ke jaringan internet. Jika perintah suara yang diinputkan benar, maka *nodeMCU* akan memproses perintah tersebut dan akan mengontrol Dimmer untuk menyalakan lampu, meredupkan dan menerangkan lampu.

3.6 Analisa Kebutuhan

3.6.1 Analisa Kebutuhan Penelitian

Alat Penelitian yang dibutuhkan dalam pembuatan tugas akhir ini yaitu berupa perangkat keras dan perangkat lunak. Adapun perangkat keras dan perangkat lunak yang dibutuhkan untuk membentuk *Perancangan Sistem Kontrol Lampu Berbasis Mikrokontroler NodeMCU Dengan Sensor Suara*, yaitu:

3.6.2 Analisa Kebutuhan Perangkat Keras

Pada analisa kebutuhan perangkat keras akan dijelaskan perangkat yang dipakai pada penelitian yaitu :

1. ESP8266 NodeMCU digunakan sebagai alat pemroses data dan alat untuk mengirimkan data yang telah diproses ke *webserver*.
2. *AC Light Dimmer Module* digunakan untuk mengontrol lampu.
3. *Smartphone* digunakan untuk pengontrolan.

3.6.3 Analisa Kebutuhan Perangkat Lunak

Pada analisa kebutuhan perangkat lunak akan dijelaskan perangkat yang dipakai pada penelitian yaitu:

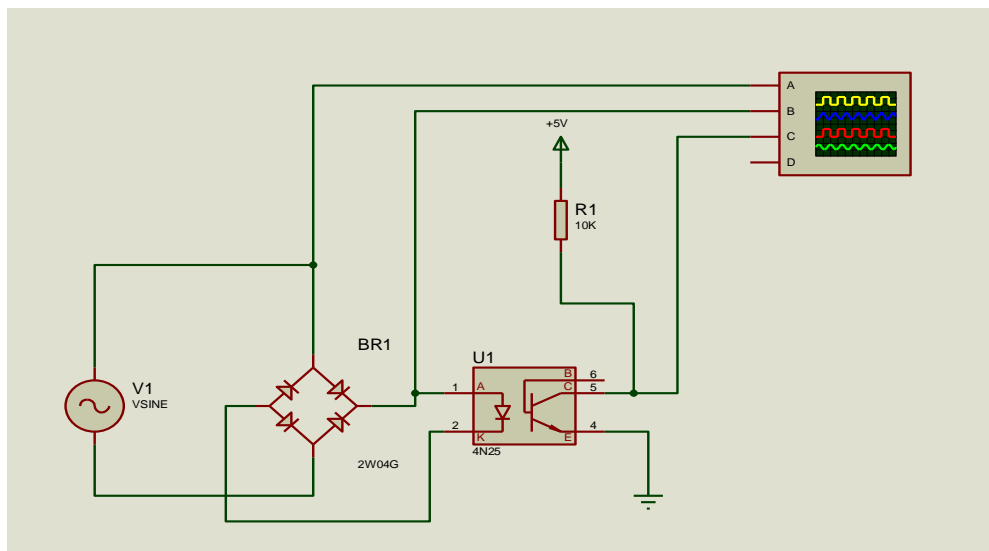
1. Arduino IDE digunakan untuk memprogram mikrokontroler, Arduino IDE yang digunakan adalah Arduino IDE versi 1.8.8
2. *APP Inventor* digunakan untuk pemrograman aplikasi *android*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian *zero crossing detector*

Zero crossing detector merupakan rangkaian elektronik yang berfungsi mendeteksi persilangan nol yang ada pada tegangan jala-jala listrik PLN. Pada rangkaian dimmer digunakan IC 4N25 sebagai *zero crossing detector (ZCD)*. Prinsip kerja IC 4N25 sama halnya dengan sebuah transistor jenis NPN yang akan aktif saat basis sebuah transistor dipicu oleh arus, namun dalam sebuah 4N25 kaki basis dipicu oleh nyala LED yang sedang dalam kondisi ON saat diberikan arus dan tegangan pada kaki anoda dan katoda. Berikut simulasi dengan menggunakan *software proteus* :



Gambar 4.1 Rangkaian pengujian IC 4N25

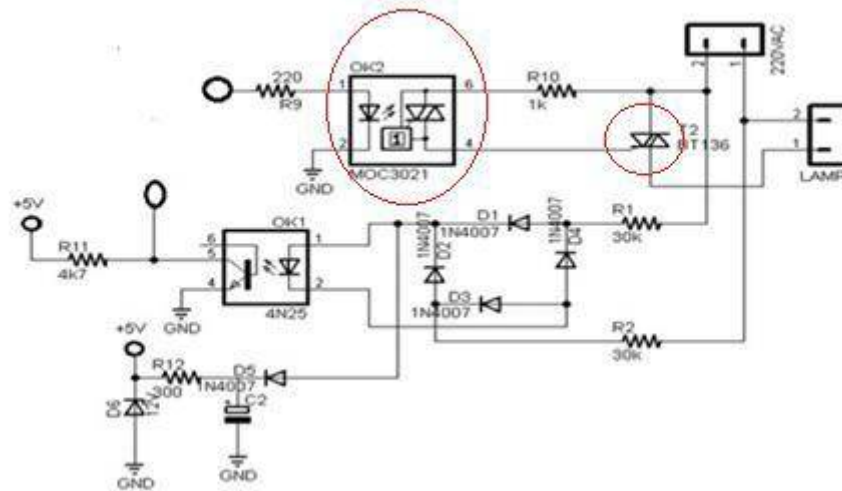
Sebuah sinyal 220V dengan frekuensi 50 Hz akan dibaca titik nol (*zero crossing*) oleh 4N25 dengan cara mengubah sinyal bolak-balik sumber AC menjadi sumber DC agar dapat menyulut LED dalam kondisi bias maju. Saat gelombang sinus

mengalami 1 periode maka oleh *diode bridge* akan diubah menjadi gelombang penuh. kemudian *peak* atau puncak tegangan akan memicu phototransistor di dalam IC 4N25 aktif. Resistor *pullup* yang terhubung akan menghasilkan gelombang puncak yang sama besarnya saat keadaan aktif sehingga menjadi indikator terjadi persilangan titik nol pada gelombang sumber 220V AC.

4.2 Pengujian dimmer lampu

Dimmer lampu digunakan untuk mengendalikan kecerahan lampu. Dimmer dapat mengendalikan lampu dengan daya yang dapat diatur, sehingga lampu menjadi samar, terang, padam. Berbeda dengan relay yang hanya bisa padam atau terang. Rangkaian *zero-cross detector* digunakan untuk menghasilkan pulsa kotak positif. Pulsa kotak ini dihasilkan ketika gelombang sinus 50 Hz, melalui titik nol.

Dari hasil rangkaian *zero crossing* dengan IC 4n25 ini akan didapatkan sinyal PWM yang tersinkronisasi dengan sinyal jala-jala 50 Hz. Sinyal ini akan masuk ke *optotriac*. *Opto triac* ini berguna untuk memisahkan jaringan tegangan DC dengan jaringan tegangan AC. Hasil PWM tadi, memicu *optotriac* lalu *optotriac* memicu TRIAC yang dihubungkan dengan beban, seperti Gambar 15. Sehingga penyalan lampu dapat dikendalikan, secara tidak langsung dari tegangan DC variabel.



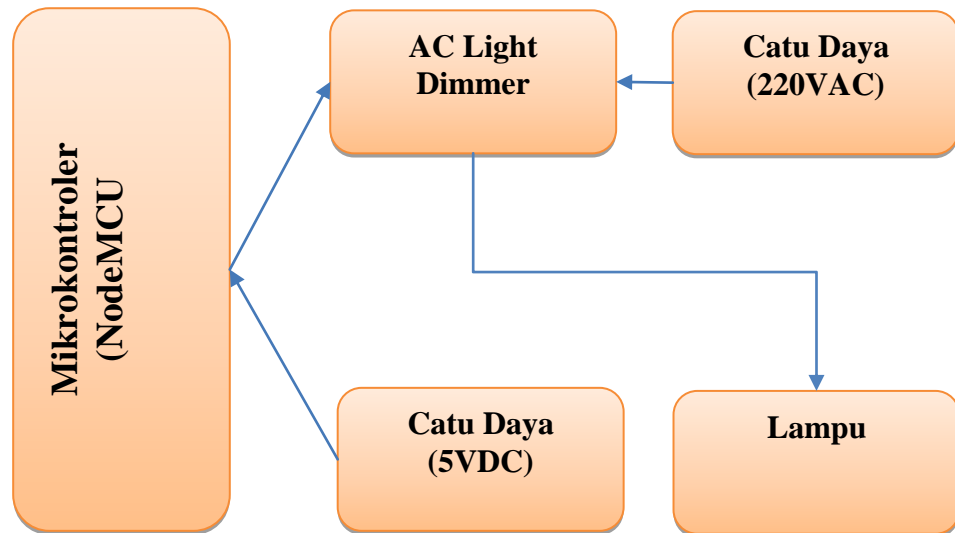
Gambar 4.2 . Hubungan opto triac dan triac

Setelah dilakukan pengujian zero crossing yang telah diuji pada percobaan diatas, maka diterapkan pada lampu yang di kendalikan melalui aplikasi android yang dibuat di App Inventor dengan perintah suara.



Gambar 4.3 Tampilan Aplikas Pengujian

4.3 Pengujian Perangkat Keras



Gambar 4.4 Blok Diagram Perangkat Keras



Gambar 4.5 Rangkaian Perangkat Keras

Perangkat keras pada tugas akhir ini terdiri dari catu daya, mikrokontroler NodeMCU, AC Light Dimmer, dan lampu. Penjelasan hasil pembuatan alat perangkat keras adalah sebagai berikut:

a. Pengukuran Catu Daya

Catu daya pada tugas akhir ini digunakan untuk menyuplai tegangan pada mikrokontroler NodeMCU dan AC Light Dimmer. Catu daya yang digunakan memiliki keluaran tegangan DC 5V. Adapun hubungan antara keluaran tegangan pada adaptor dengan waktu bisa dilihat pada tabel dibawah ini:

Tabel 4.1 Pengujian Rangkaian Catu Daya 5V

No	Waktu (Menit)	Tegangan Keluaran Adaptor (V)
1	(0-5)	(5.12 ± 0.01)
2	(5-10)	(5.11 ± 0.01)
3	(10-15)	(5.11 ± 0.01)
4	(15-20)	(5.12 ± 0.01)
5	(20-25)	(5.11 ± 0.01)
6	(25-30)	(5.12 ± 0.01)

Dari hasil pengujian (Tabel) menunjukkan bahwa tegangan keluaran dari rangkaian catu daya hampir sesuai dengan yang dibutuhkan, yaitu sekitar +5V. Dari tabel diatas terlihat juga bahwa tegangan yang dikeluarkan oleh rangkaian catu daya ini selama percobaan 30 menit keadaannya cenderung stabil.

b. Pengukuran Mikrokontroler dan AC Light Dimmer

Dimmer yang digunakan pada tugas akhir adalah AC Light Dimmer. Pada mikrokontroler, pin D5, pin D6, pin D7, dan pin D8 digunakan sebagai output yang terhubung ke AC Light Dimmer. Pin-pin pada mikrokontroler

ini menggunakan tegangan 3,3V. Pengukuran dilakukan dengan mengukur tegangan pin D5, pin D6, pin D7, dan pin D8 saat Dimmer dalam keadaan OFF, ON Terang dan Redup, dengan tabel pengukuran diperlihatkan pada tabel 4.2 dibawah ini:

Table 4.2 Pengukuran Tegangan LOAD AC Light Dimmer

No	Lampu	Tegangan LOAD AC light Dimmer
1	ON	255 VAC
2	OFF	0 VAC
3	REDUP	30 VAC
4	TERANG	250 VAC

Dari hasil pengujian (Tabel 4) menunjukkan bahwa tegangan keluaran dari LOAD AC Light Dimmer yaitu sekitar 255 VAC saat kondisi keluaran ON, 0 VAC saat kondisi mati, 30 VAC saat kondisi saar dan sekitar 250 VAC saat kondisi terang.

4.4 Pengujian Perangkat Lunak

Pengujian perangkat lunak pada tugas akhir ini meliputi pengujian sistem Wi-Fi pada alat, dan pengujian alat dan sistem terhadap respon perintah suara yang diberikan melalui aplikasi yang dibuat di App Inventor.

a. Pengujian Wi-Fi Manager pada Sistem

Seperti dijabarkan pada bab 3, pada saat sistem dinyalakan, mikrokontroler akan melakukan inisialisasi dan mencoba melakukan koneksi ke akses poin yang terakhir terhubung. Apabila koneksi berhasil terhubung, mode Wi-Fi pada alat akan menjadi station, jika koneksi gagal

mode Wi-Fi pada alat akan menjadi access point dengan nama SSID: “userinternet”, dan PASSWORD: “password123”.

Pengujian dilakukan dengan menggunakan program Arduino IDE pada komputer, menggunakan fitur Serial Monitor pada program. Sebelumnya mikrokontroler dihubungkan dengan komputer dengan kabel micro usb. Wi-Fi akses poin internet pada pengujian ini menggunakan Portabel Wi-Fi hotspot smartphone android. Pada pengujian pertama Portabel Wi-Fi hotspot dimatikan, tujuannya untuk menguji mode access point pada alat.

Proses yang terjadi pada sistem pada pengujian ini bisa dilihat di Serial Monitor pada gambar diatas. Setelah mode Wi-Fi pada alat menjadi access point, langkah selanjutnya adalah menyambungkan ke SSID alat, pada pengujian ini digunakan komputer untuk melakukan koneksinya. Setelah komputer terhubung ke access point pada alat, maka akan diarahkan ke portal Wi-Fi manager pada browser dengan IP Address “192.168.4.1”.

Program yang digunakan untuk mendeteksi sinyal WIFI dan koneksi otomatis dapat dilihat pada bagian bawah :

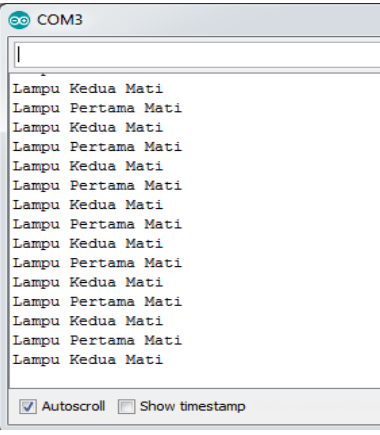

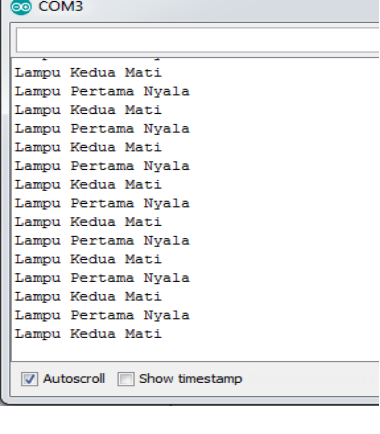

```
WiFi.disconnect();
delay(3000);
WiFi.begin("Redmi", "1234567890123");
while (!(WiFi.status() == WL_CONNECTED)){
  delay(300);
}
Serial.println("Tersambung");
}
```

b. Pengujian dengan Perintah Suara Aplikasi AC Light Dimmer

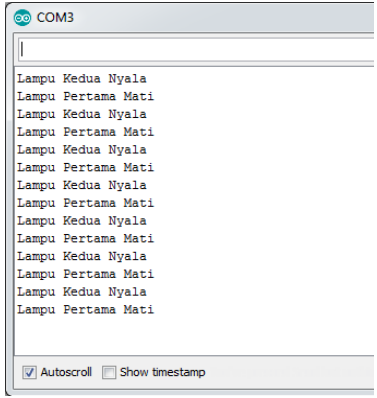

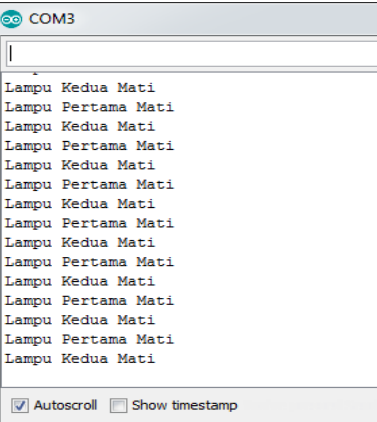

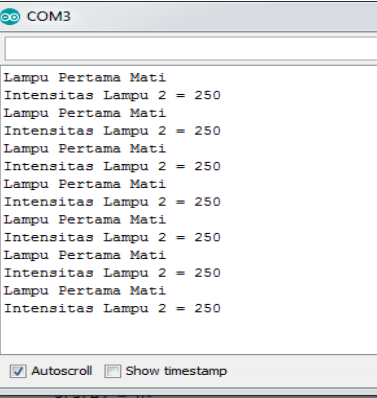

Sistem yang dibuat ini bisa memungkinkan user untuk menyalakan dan mematikan lampu yang terhubung pada Dimmer melalui perintah suara dengan aplikasi AC Light Dimmer yang dibuat di App Inventor pada *smartphone*. Perintah suara menggunakan Indonesia yang mana speech recognition sudah mendukung bahasa Indonesia. Input perintah suara yang diberikan untuk mengendalikan dimmer ada 6 perintah suara, antara lain:

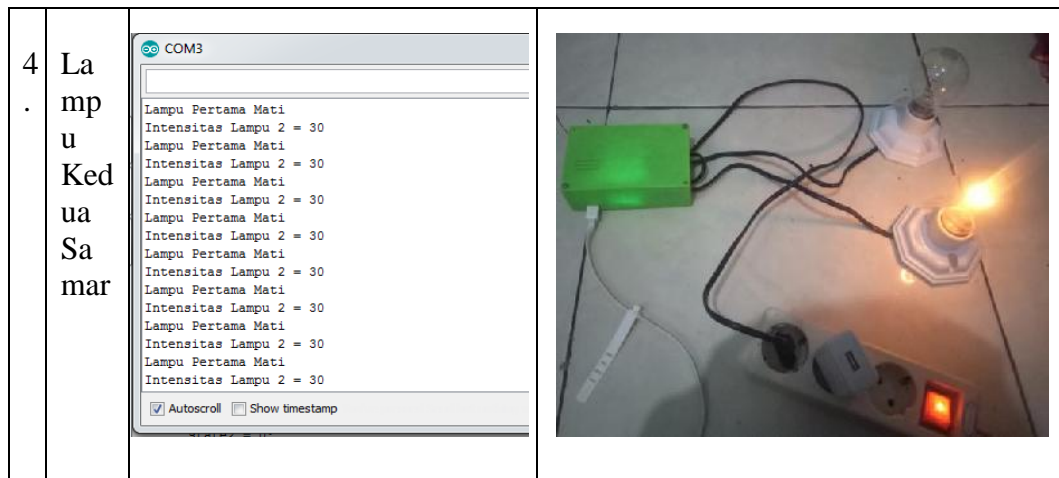
- 1) Perintah suara untuk mengontrol lampu nomor 1.

Tabel 4.3 : Perintah dan hasil

No	Perintah	Serial Monitor	Hasil
1	Lampu Pertama Mati		
2	Lampu Pertama Nyala		

2) Perintah suara untuk mengontrol lampu nomor 2

No	Perintah	Serial Monitor	Hasil
1	Lampu Kedua Nyala	 <pre> COM3 Lampu Kedua Nyala Lampu Pertama Mati Lampu Kedua Nyala Lampu Pertama Mati Lampu Kedua Nyala Lampu Pertama Mati Lampu Kedua Nyala Lampu Pertama Mati Lampu Kedua Nyala Lampu Pertama Mati Lampu Kedua Nyala Lampu Pertama Mati Lampu Kedua Nyala Lampu Pertama Mati Lampu Kedua Nyala Lampu Pertama Mati </pre>	
2	Lampu Kedua Mati	 <pre> COM3 Lampu Kedua Mati Lampu Pertama Mati Lampu Kedua Mati Lampu Pertama Mati Lampu Kedua Mati Lampu Pertama Mati Lampu Kedua Mati Lampu Pertama Mati Lampu Kedua Mati Lampu Pertama Mati Lampu Kedua Mati Lampu Pertama Mati Lampu Kedua Mati Lampu Pertama Mati Lampu Kedua Mati Lampu Pertama Mati </pre>	
3	Lampu Kedua Terang	 <pre> COM3 Lampu Pertama Mati Intensitas Lampu 2 = 250 Lampu Pertama Mati Intensitas Lampu 2 = 250 Lampu Pertama Mati Intensitas Lampu 2 = 250 Lampu Pertama Mati Intensitas Lampu 2 = 250 Lampu Pertama Mati Intensitas Lampu 2 = 250 Lampu Pertama Mati Intensitas Lampu 2 = 250 Lampu Pertama Mati Intensitas Lampu 2 = 250 Lampu Pertama Mati Intensitas Lampu 2 = 250 </pre>	



Selanjutnya untuk setiap perintah suara yang diberikan, Aplikasi AC Light Dimmer akan memberikan respon sebagai berikut:

- a) Respon untuk perintah suara menyalakan lampu nomor 1
“Lampu satu telah dinyalakan”
- b) Respon untuk perintah suara menyalakan lampu nomor 2
“Lampu dua telah dinyalakan”
- c) Respon untuk perintah suara mematikan lampu nomor 1
“Lampu satu telah dimatikan”
- d) Respon untuk perintah suara mematikan lampu nomor 2
“Lampu dua telah dimatikan”
- e) Respon untuk perintah suara menerangkan lampu nomor 2
“Lampu dua terang”
- f) Respon untuk perintah suara meredupkan lampu nomor 2
“Lampu dua samar”

Untuk mendapatkan hasil percobaan yang lebih akurat, pengujian dilakukan dengan 2 buah smartphone yang berbeda, dan dengan 2 orang

untuk input perintah suara. Spesifikasi smartphone yang digunakan dalam pengujian ini adalah sebagai berikut:

a) Redmi

- CPU: Mediatek MT6750 (28 nm)
- RAM: 4GB
- CPU Arch: Octa-core (4x1.5 GHz Cortex-A53 & 4x1.0 GHz Cortex-A53)
- OS: Stock ROM
- Android: 6.0

b) Oppo A71

- CPU: Qualcomm SDM660 Snapdragon 660 (14 nm)
- RAM: 2 GB
- CPU Arch: EightCore (4x2.2 GHz Kryo 260 & 4x1.8 GHz Kryo 260)
- OS: v3.1
- Android: 8.1

Perintah suara yang masuk pada *smartphone* akan diubah menjadi kode-kode yang unik yang diterjemahkan langsung oleh web server ThingSpeak.com. kode-kode yang unik dimaksudkan agar setiap pengguna *library* ThingSpeak dapat membuat perintahnya sendiri dan tidak bersinggungan dengan perintah atau proyek yang dibuat orang lain. Perintah-perintah tersebut akan dikirimkan ke nodeMCU yang dapat dilihat pada program seperti dibawah :

```
int val1 = (ThingSpeak.readIntField(659329, 1, "PG4EDYV19ZTU4CNJ"));
int val2 = (ThingSpeak.readIntField(659329, 2, "PG4EDYV19ZTU4CNJ"));
```


terdapat dua kode pada program diatas yang dapat digunakan pada lampu 1 dan lampu 2. Val adalah sebuah variabel yang dibentuk dan memiliki nilai integer (bilangan bulat). Yang sebenarnya adalah, pada aplikasi yang dibuat pada app Invertor, setiap perintah suara yang didapat akan diterjemahkan ke dalam angka dalam bilangan bulat. Angka ini nantinya bertujuan akan menjadi nilai intensitas pada lampu. Jika ditemukan perintah “Lampu pertama mati”, aplikasi akan menerjemahkan perintah suara tersebut menjadi angka “0” dan akan dikirimkan ke nodeMCU yang akan menjadi nilai intensitas dari lampu. Begitu juga seterusnya. Program untuk menerjemahkan perintah tersebut dapat dilihat pada bagian dibawah :

```

if(val1 == 300){
  digitalWrite(pwmPin1, 1);
  tarBrightness1 = 255;
  state1 = 1;
  Serial.println("Lampu Pertama Nyala");
}
if(val2 == 300){
  digitalWrite(pwmPin2, 1);
  tarBrightness2 = 255;
  state2 = 1;
  Serial.println("Lampu Kedua Nyala");
}
if(val1 == 400){
  digitalWrite(pwmPin1, 0);
  state1 = 0;
  tarBrightness1 = 0;
  Serial.println("Lampu Pertama Mati");
}
if(val2 == 400){
  digitalWrite(pwmPin2, 0);
  state2 = 0;
  tarBrightness2 = 0;
  Serial.println("Lampu Kedua Mati");
}
if(val1 > 0 && val1 < 255){
  digitalWrite(pwmPin1, 1);
  state1 = 1;
  tarBrightness1 = val1;
  Serial.print("Intensitas Lampu 1 = ");

```

```

        Serial.println(tarBrightness1);
    }
    if(val2 > 0 && val2 < 255){
        digitalWrite(pwmPin2, 1);
        state2 = 1;
        tarBrightness2 = val2;
        Serial.print("Intensitas Lampu 2 = ");
        Serial.println(tarBrightness2);
    }
}

```

Intensitas lampu yang dapat dibaca dan diproses oleh nodeMCU bernilai dari angka 0 hingga 255. Jika diperhatikan pada program, dapat ditemukan nilai val yang melebihi dari angka 255. Ini dimaksudkan untuk pengalih dari perintah menyala dan padam dikarenakan *library* yang didapatkan oleh pembuat sebelumnya tidak dapat mengalihkan intensitas untuk angka “0” ke intensitas di atasnya dalam artian akan terhenti diangka “0” jika didapati nilai intensitas pada angka 0. Untuk itu perlu dilakukan modifikasi agar lampu dapat menyala secara bebas dari redup ke terang, dari terang ke padam dan seterusnya.

Untuk memastikan lampu dapat menyala terang ke redup dengan pergantian intensitas yang halus. Digunakan program seperti terlihat pada bagian berikut :

```

void dimTimerISR1() {
    if (fade1 == 1) {
        if (curBrightness1 > tarBrightness1 || (state1 == 0 && curBrightness1 > 0)) {
            --curBrightness1;
        }
        else if (curBrightness1 < tarBrightness1 && state1 == 1 && curBrightness1 < 255)
        {
            ++curBrightness1;
        }
    }
    else {
        if (state1 == 1) {
            curBrightness1 = tarBrightness1;
        }
        else {
            curBrightness1 = 0;
        }
    }
}

```

```

    }

    if (curBrightness1 == 0) {
        state1 = 0;
        digitalWrite(pwmPin1, 0);
    }
    else if (curBrightness1 == 255) {
        state1 = 1;
        digitalWrite(pwmPin1, 1);
    }
    else {
        digitalWrite(pwmPin1, 1);
    }

    zcState1 = 0;
}

void zcDetectISR1() {
    if (zcState1 == 0) {
        zcState1 = 1;

        if (curBrightness1 < 255 && curBrightness1 > 0) {
            digitalWrite(pwmPin1, 0);

            int dimDelay1 = 30 * (255 - curBrightness1) + 400;//400
            hw_timer_arm(dimDelay1);
        }
    }
}
}

```

Program diatas harus mengacu pada library “HW_Timer.h” agar fungsi dari variabel-variabel yang dibentuk dapat digunakan dengan baik.

4.5 Pengiriman Nilai Data ke ThingSpeak

Pengujian pada ThingSpeak server dilakukan untuk mengetahui apakah nilai field pada channel ThingSpeak dapat diubah sesuai dengan nilai data yang dikirimkan melalui protokol HTTP. Untuk pengujian ini, pengiriman nilai data menggunakan aplikasi android. Nilai data yang dikirimkan pertama adalah 0 untuk mematikan lampu, 255 untuk menghidupkan lampu, 120 untuk meredupkan lampu dan kemudian 250 untuk menerangkan lampu.

Pada aplikasi android diisikan Write API Key agar dapat menulis ke dalam channel (disensor untuk kepentingan privasi) kemudian nomor field yang dituju

(dalam pengujian ini menggunakan field ke 2) lalu diikuti dengan nilai yang akan dikirimkan. Setelah itu pengujian dilanjutkan dengan melakukan pengiriman nilai data beberapa kali untuk mengetahui respon ThingSpeak terhadap pengiriman data yang dilakukan terus menerus. Pengujian dilakukan setiap interval waktu 3 detik selama 30 detik dan hasil pengujian Perintah Thingspeak Dari Android dapat dilihat pada tabel 4.4

Tabel 4.4 Hasil pengujian pengiriman nilai data ke ThingSpeak

Pengiriman ke Thingspeak		
Detik	Hasil	Waktu (s)
0	Berhasil	1,70
3	Berhasil	2,00
6	Berhasil	1,90
9	Berhasil	1,80
12	Berhasil	1,50
15	Gagal	1,60
18	Berhasil	1,90
21	Berhasil	2,10
24	Gagal	1,50
27	Berhasil	1,70
30	Berhasil	1,80
Waktu Rata-Rata		1,77

Dari hasil pengujian bisa disimpulkan bahwa pengiriman dan perubahan data pada *ThingSpeak* melalui protokol HTTP dapat dilakukan. Dapat dilihat pula waktu rata-rata yang diperlukan untuk setiap pengiriman data pada tabel di atas adalah 1,77 detik.

4.6 Pembacaan Data Thingspeak Pada Modul NodeMcu

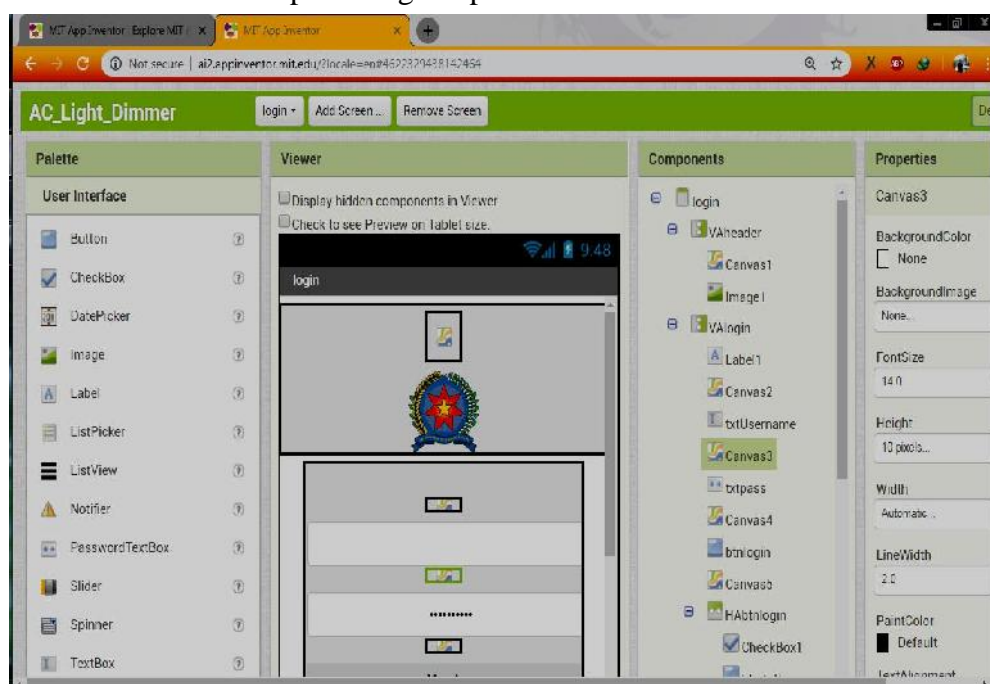
Pengujian ini dilakukan untuk mengetahui apakah mikrokontroler yang digunakan dapat melakukan pembacaan terhadap tag yang akan digunakan pada perancangan. Pada dasarnya. Namun, untuk melakukan penulisan terhadap tag diperlukan sebuah aplikasi yang dapat membuat *smartphone* dapat berfungsi

sebagai NFC *writer*. Banyak pilihan aplikasi yang beredar pada pasar aplikasi Google yang dapat digunakan untuk melakukan penulisan tag untuk sistem operasi Android yang mana pada pengujian ini penulis menggunakan aplikasi NFC *Tools*. Pada tugas akhir ini aplikasi android dibuat dengan App Inventor yang didesain oleh penulis sendiri.

4.7 Pembuatan Aplikasi Android

Aplikasi *virtual button* dibutuhkan sebagai media masukan sentuhan melalui smartphone yang dibuat menggunakan app inventor. Berikut merupakan langkah-langkah dalam pembuatan aplikasi *virtual button*:

a. Mendesain Tampilan Login Aplikasi



Gambar 4.6 Tampilan Login Aplikasi

Gambar 20 menunjukkan window pembuatan screen login aplikasimenggunakan app inventor yang berisi username dan password aplikasididesainsebagai hak akses dan menjaga keamanan aplikasi dari

pengguna lain. Untuk *Source Code* blok pada menu Login adalah sebagai berikut :



Gambar 4.7 *Source Code* Blok Menu Login

b. Mendesain tampilan Menu Pemilihan



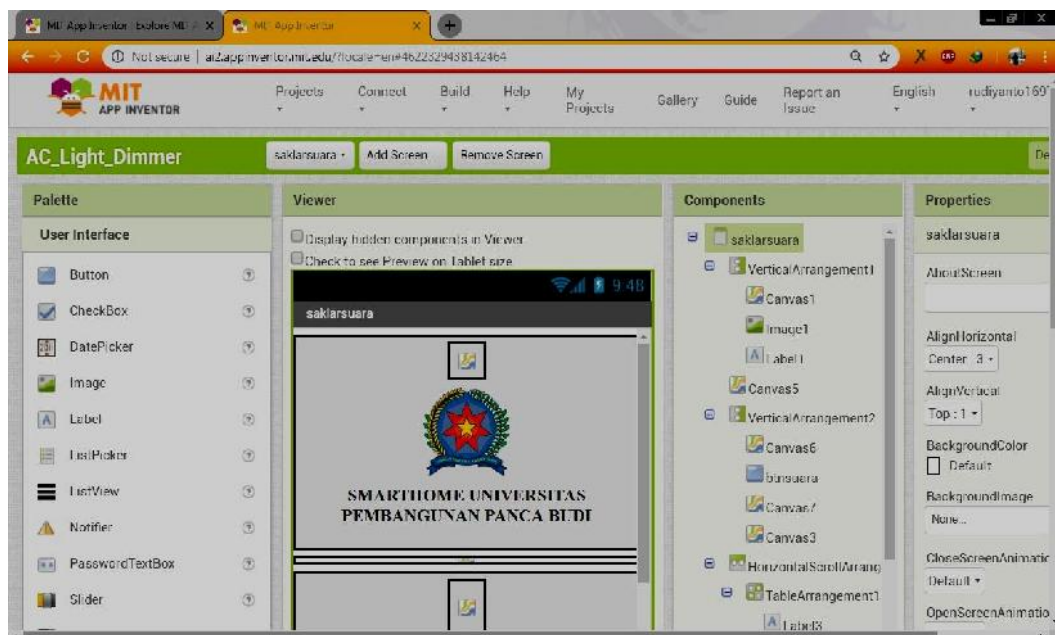
Gambar 4.8 Tampilan Menu Aplikasi

pilih saklar suara untuk membuka menu perintah suara. Untuk *Source Code* blok tampilan menu dapat dilihat pada gambar berikut :



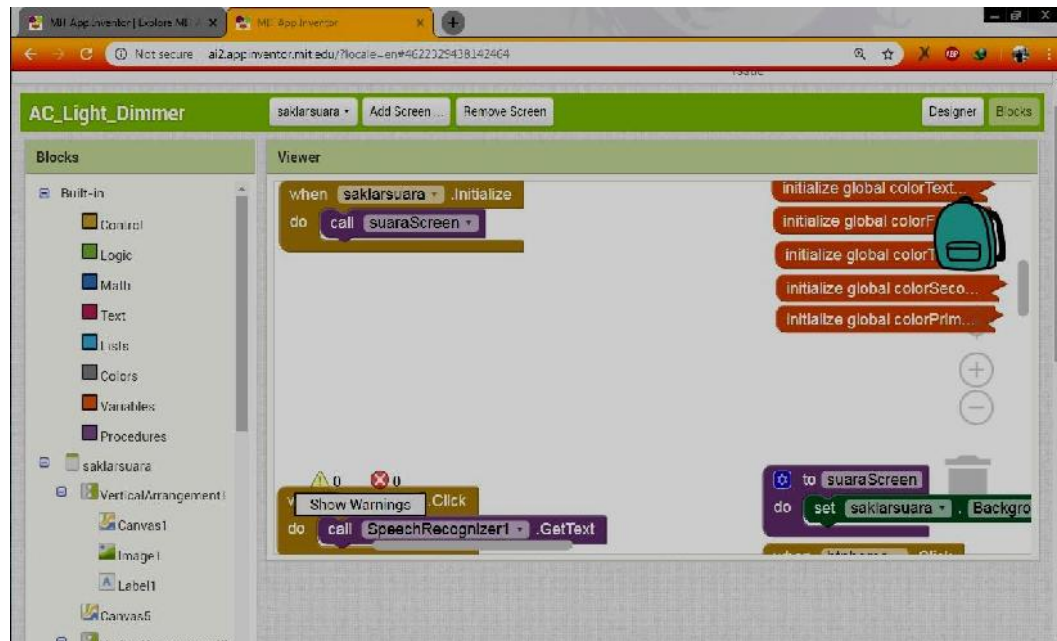
Gambar 4.9 *Source Code* Tampilan Menu

c. Tampilan Menu Perintah Suara



Gambar 4.10 Tampilan Menu Perintah Suara

Gambar 24 adalah menu perintah suara untuk mengontrol lampu yang didalamnya terdapat button Speech Recognition untuk menyambungkan alat dengan aplikasi. Untuk *Source Code* blok perintah suara dapat dilihat pada gambar berikut :



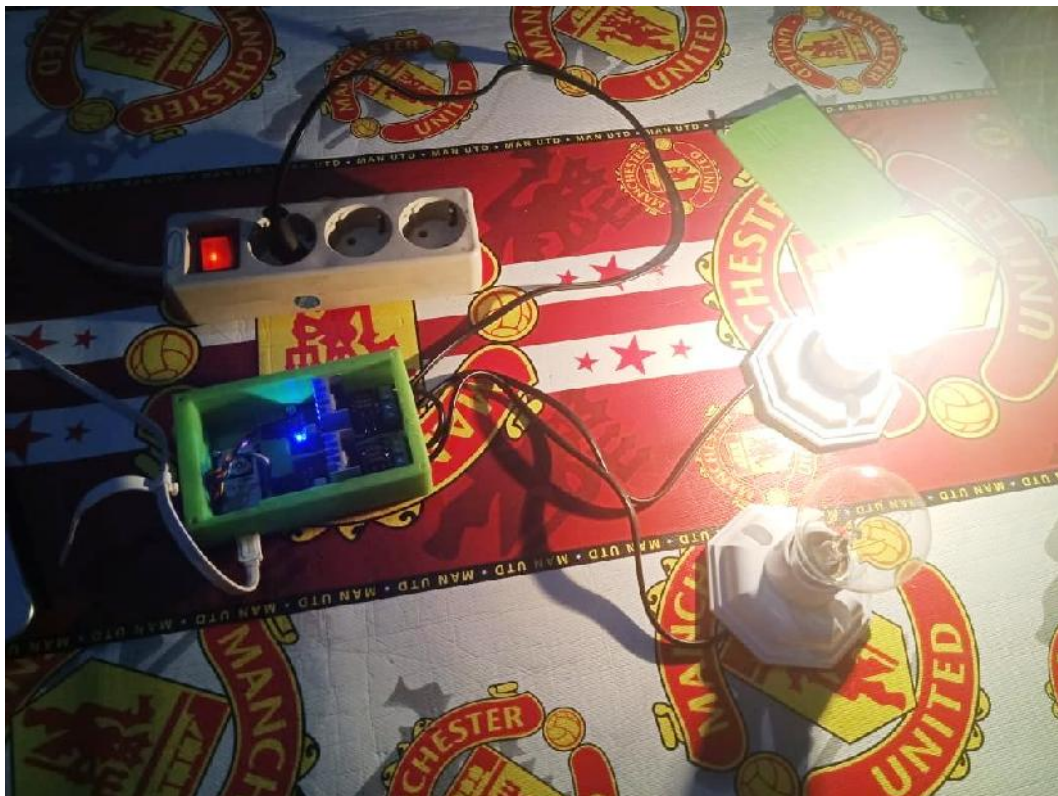
Gambar 4.11 *Source Code* Blok Perintah Suara

4.8 Pengujian Secara Keseluruhan

Pengujian secara keseluruhan dilakukan untuk mendapatkan hasil akhir yang akan menentukan kinerja alat yang dirancang secara keseluruhan. Setelah semua program awal baik *hardware* maupun *software* di uji coba dan didapatkan hasil yang baik, maka langkah selanjutnya adalah dengan menyatukan keseluruhan program untuk dapat dilakukan pengujian secara keseluruhan.

Pengujian dilakukan menggunakan dua buah lampu pijar agar dapat dibedakan intensitas lampu redup dan terang nantinya. Tentunya lampu LED 220 volt yang banyak digunakan saat ini telah menggunakan teknologi yang lebih baik

untuk tidak terlalu terpengaruh pada tegangan listrik yang lemah karena sesungguhnya lampu LED yang digunakan tidak membutuhkan listrik dengan daya yang besar untuk dapat menyala. Untuk itu pengujian dengan menggunakan lampu LED tidak akan mendapatkan hasil yang baik untuk mengetahui terang redupnya lampu.



Gambar 4.12 Percobaan Secara keseluruhan

BAB V

PENUTUP

5.1 Kesimpulan

- a. Pembuatan software virtual button menggunakan App Inventor untuk mendesain dan memprogram aplikasi secara online. Program yang digunakan untuk menjalankan mikrokontroler NodeMCU menggunakan Arduino IDE dan pembuatan *source code* program menggunakan bahasa C
- b. Pembuatan hardware menggunakan NodeMCU dan AC Light Dimmer serta lampu pijar. Jangkauan jarak pengontrolan adalah tidak terbatas asal mikrokontroler NodeMCU dan smartphone tersambung dengan internet.
- c. Hasil untuk kerja pengontrolan lampu menggunakan smartphone berbasis NodeMCU sudah bekerja sesuai dengan fungsinya dengan keberhasilan 100%.

5.2 Saran

Berdasarkan alat yang sudah dibuat maka penulis menemukan dan menyarankan agar ketika alat ini benar-benar diterapkan maka agar lebih berhati-hati dalam penggunaan dikarenakan kerentanan alat ini dapat dikontrol oleh user manapun walaupun ssid dan password berbeda antara alat dan aplikasi android. Untuk penelitian selanjutnya ada baiknya dengan menggunakan *arduino board* yang telah dilengkapi ESP8266 sebagai pemancar sinyal WIFI agar penggunaan yang lebih *flexible* terhadap dimmer jika ingin menggunakan lebih banyak lampu yang akan dikontrol. Dikarenakan *output* yang dikeluarkan oleh *nodeMCU board* berada pada level 3.3 volt yang mengakibatkan lemahnya *output* logika 1 yang

akan diterima oleh dimmer. Mengakibatkan *board* NodemCU sulit untuk mengeluarkan sinyal PWM yang dapat mengontrol terang-redupnya lampu yang dikontrol melalui dimmer. Untuk itu penggunaan arduino *board* dapat menjadi solusi untuk pengembangan selanjutnya dan juga dukungan library pada arduino *board* yang lebih flexible untuk mendapatkan banyak sinyal PWM pada output-outputnya.

DAFTAR PUSTAKA

- Akbar, A. (2018). Pembangunan Model Electronic Government Pemerintahan Desa Menuju Smart Desa. *Jurnal Teknik dan Informatika*, 5(1), 1-5.
- appinventor. (2019). app inventor. Retrieved February 23, 2019, from <http://aai2.appinventor.mit.edu>
- Apri, J. (2015). Internet of Things, Sejarah, Teknologi Dan Penerapannya : Review. *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3).
- Baidhowi, A., & Neforawati, I. (2017). Rancang Bangun Alat Konrtol Smart room melalui Perintah Suara Berbasis Android. *Jurnal Politeknik Negeri Surabaya*.
- Dimas, A., Sadewo, B., Widasari, E. R., & Muttaqin, A. (2017). Perancangan Pengendali Rumah menggunakan Smartphone Android dengan Konektivitas Bluetooth. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1(5), 415–425.
- Embeddednesia. (2017). tutorial nodemcu pertemuan pertama. Retrieved from <https://embeddednesia.com/v1/tutorial-nodemcu-pertemuan-pertama/>
- haris abdul. (2014). *sistem kendali pembatas pemakaian energi listrik berbasis web*.
- kurniawan adris. (2014). RANCANG BANGUN SISTEM PENGENDALIAN SUHU PADA
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365.
- Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, 9(2), 103-122.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. *Journal of Saintech Transfer*, 1(1), 27-32.
- Herdianto, H. (2018). Perancangan Smart Home dengan Konsep Internet of Things (IoT) Berbasis Smartphone. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).

- INKUBATOR BAYI BERBASIS Andris Kurniawan Bambang Suprianto Abstrak. *Jurnal Universitas Negeri Surabaya, Vol. 07 No, 22–223.*
- Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. *Jurnal Teknologi Informasi dan Pendidikan, 11(1), 1-6.*
- Kurnia, D. (2017). Analisis QoS Pada Pembagian Bandwidth Dengan Metode Layer 7 Protocol, PCQ, HTB Dan Hotspot Di SMK Swasta Al-Washliyah Pasar Senen. *CESS (Journal of Computer Engineering, System and Science), 2(2), 102-111.*
- Kurnia, D., Dafitri, H., & Siahaan, A. P. U. (2017). RSA 32-bit Implementation Technique. *Int. J. Recent Trends Eng. Res, 3(7), 279-284.*
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol, 38(7), 380-383.*
- Marlina, L., Putera, A., Siahaan, U., Kurniawan, H., & Sulistianingsih, I. (2017). Data Compression Using Elias Delta Code. *Int. J. Recent Trends Eng. Res, 3(8), 210-217.*
- Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research, 3(8), 196-201.*
- Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. *Int. J. Eng. Technol., 7(2.13), 345-347.*
- Ramadhani, S., Suherman, S., Melvasari, M., & Herdianto, H. (2018). Perancangan Teks Berjalan Online Sebagai Media Informasi Nelayan. *Jurnal Ilmiah Core IT: Community Research Information Technology, 6(2).*
- Rasimah, N. (2014). *Kecerdasan Buatan. 211–235.*
- Ruwaida, D., & Kurnia, D. (2018). Rancang Bangun File Transfer Protocol (FTP) dengan Pengamanan Open SSL pada Jaringan VPN Mikrotik di SMK Dwiwarna. *CESS (Journal of Computer Engineering, System and Science), 3(1), 45-49.*
- Samsugi, S., & Kastutara, D. (2018). ARDUINO DAN MODUL WIFI ESP8266 SEBAGAI MEDIA KENDALI JARAK JAUH DENGAN ANTARMUKA BERBASIS ANDROID. *Jurnal Teknoinfo Vol.12 No.1 Pp: 23-17. ISSN 1693-0010, 12(1), 23–27.*

- Setiawan, E. T. (2016). Pengendalian lampu rumah berbasis mikrontroler arduino menggunakan smartphone android. *TI-Atma STMIK Atma Luhur Pangkalpinang*, 1–8.
- Sinauarduino. (2016). mengenal software arduino IDE. Retrieved from <https://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>
- Yuliansyah, H. (2016). Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis Rest Architecture. *Jurnal Rekayasa Dan Teknologi Elektro, Vol.10 No.*