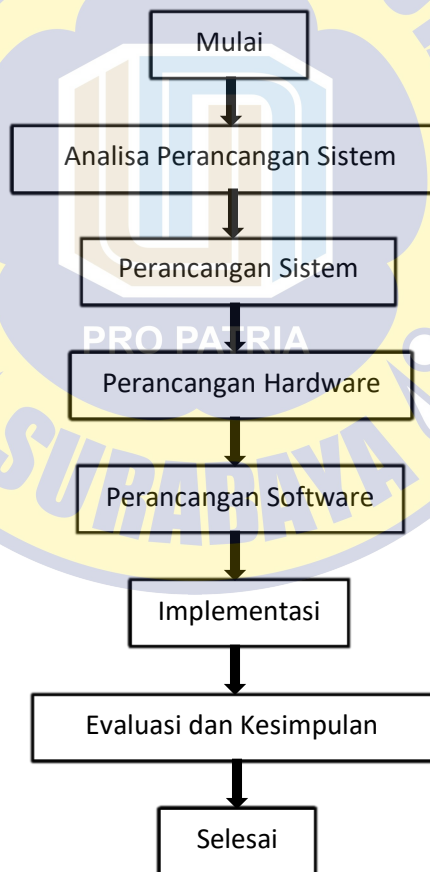


BAB III

METODOLOGI PENELITIAN

Penelitian “Record Activity Sepeda Motor Berbasis IoT” ini menggunakan metodologi antara lain meliputi studi literature, analisa permasalahan, perancangan *hardware*, penulisan program, implementasi sistem, pengujian sistem, evaluasi hasil serta dokumentasi penelitian. Adapun urutan metodologi dalam penelitian ini dapat ditampilkan dalam diagram alur berikut:



Gambar 3.1 Metode Diagram Blok

3.1 Analisa Perancangan Sistem

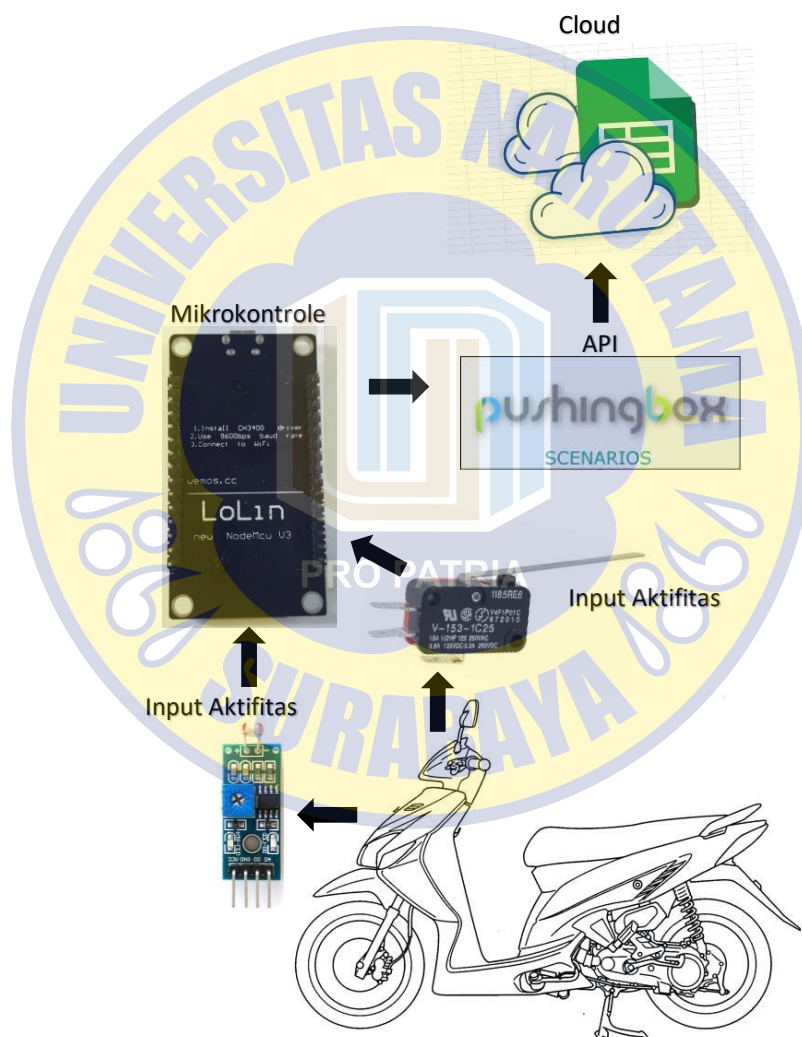
Pada tahap ini merupakan tahap yang meliputi pencarian dasar-dasar teori yang dibutuhkan dalam penelitian dengan mencari referensi dari penelitian terdahulu. Dimulai dari dasar rangkaian elektronik, komponen-komponen pendukung, bahasa pemrograman yang akan digunakan serta teori-teori pendukung lain yang terkait dengan penelitian. Tahap ini dilakukan mulai dengan membaca langsung dari beberapa jurnal terdahulu, e-book maupun artikel dari laman internet. Setelah beberapa sumber referensi didapat kemudian dipelajari, dirangkum dan ditulis kembali sesuai kebutuhan dalam penelitian.

Dalam penelitian ini, lebih terfokus pada bagaimana *record* data aktivitas sepeda motor dilihat dan didapat dengan menggunakan masukan dari sensor LDR, dan Limit Switch kemudian diproses oleh mikrokontroler NodeMCU ESP8266 dan keluarannya berupa data excel yang dapat diunggah ke aplikasi Google Spreadsheet sehingga dapat diakses secara *online*.

3.2 Perancangan Sistem

Pada perancangan alat di tahap ini diperlukan beberapa sensor untuk mendeteksi aktivitas sepeda motor seperti aktivitas pengereman (rem depan maupun rem belakang), aktivitas penggunaan lampu sein kanan maupun sein kiri, dan yang terakhir adalah penggunaan lampu depan dan lampu belakang sepeda motor. Berdasarkan beberapa jurnal maupun artikel yang telah dijadikan sebagai

rujukan, untuk mendeteksi penggunaan rem diperlukan sensor yang dapat menerima masukan berupa sentuhan atau tekanan sampai batas tertentu. Selanjutnya untuk mendeteksi aktivitas penggunaan lampu sein, lampu depan maupun lampu belakang, diperlukan sensor yang dapat menerima masukan berupa perubahan intensitas cahaya.



Gambar 3.2 Alur Sistem

Setelah semua sensor terpenuhi, selanjutnya diperlukan mikrokontroller sebagai pengolah masukan dari sensor-sensor tersebut. Kemudian data yang telah

diterima oleh mikrokontroller, agar dapat diakses secara online diperlukan modul wifi yang terintegrasi dengan mikrokontroller sebagai pengirim data secara online dan protokol API sebagai sebagai pengatur skenario upload serta Cloud Service sebagai penyimpanannya seperti pada gambar 3.2 di atas.

Adapun *hardware* dan *software* yang akan digunakan untuk perancangan dalam penelitian ini, antara lain:

1. Perangkat Keras (*Hardware*)

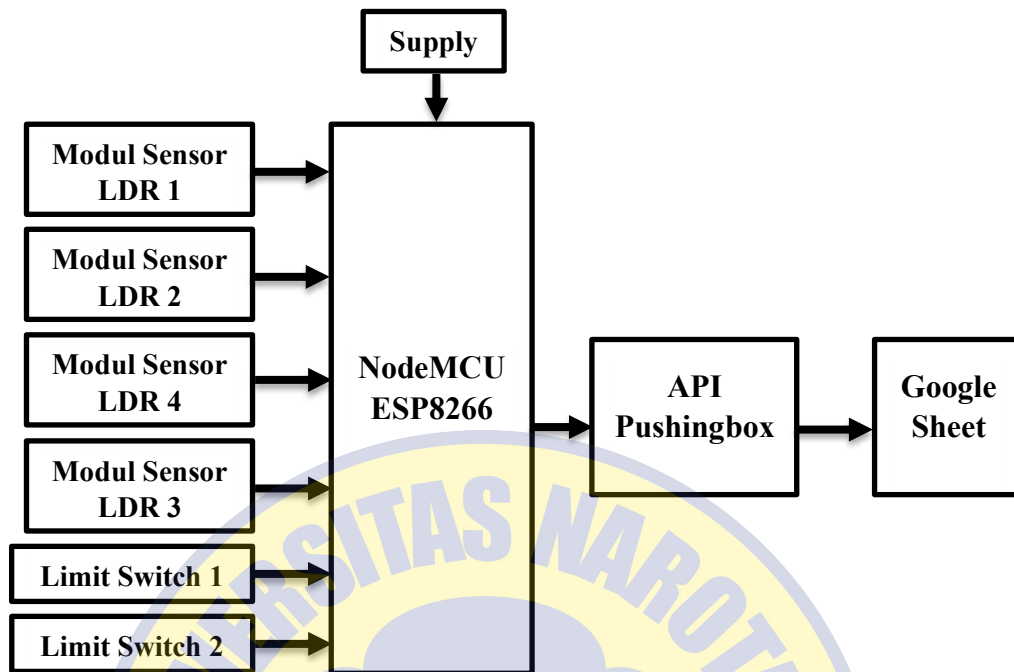
Perangkat keras yang digunakan adalah:

- NodeMCU ESP8266
- Sensor LDR
- Limit Switch
- LED
- Resistor

2. Perangkat lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian ini adalah:

- Sistem Operasi *Windows 10*
- Bahasa Pemrograman *Arduino IDE*
- Google Sheet
- API Pushingbox



Gambar 3.3 Blok Diagram Sistem Record Activity

A. Blok Input

Pada blok Input, yaitu modul sensor LDR berjumlah 4 buah sebagai pendeteksi aktivitas lampu depan, belakang, sein kiri, sein kanan dan 2 buah Limit Switch sebagai indikator penggunaan rem depan dan belakang.

B. Blok Proses

Dalam blok proses, terdapat mikrokontroller NodeMCU ESP8266 yang berperan untuk memproses data yang didapat dari sensor LDR dan Limit Switch dan API Pushingbox sebagai pemberi URL ke NodeMCU agar *output* data dapat tampil dan tersimpan di Google Spreadsheet.

C. Blok Output

Dalam blok *output*, terdapat tampilan data berupa tabel yaitu Google Sheet untuk *output* secara *online*.

Adapun tahap dalam pembuatan Record Activity Sepeda Motor Berbasis IoT adalah sebagai berikut :

- Merangkai hardware yang dibutuhkan. Untuk membuat alat Record Activity Sepeda Motor Berbasis IoT yang terdiri dari NodeMCU ESP8266, Sensor LDR dan Limit Switch.
- Memasukkan sumber kode ke dalam Arduino IDE, dalam hal ini penulis membuat code pada NodeMCU ESP8266 sekaligus menguji untuk memeriksa apakah ada *bug code* maupun sensor-sensor sudah berfungsi dengan baik.
- Memasukkan sumber kode ke dalam NodeMCU ESP8266, dalam hal ini untuk memeriksa apakah NodeMCU ESP8266 sudah berfungsi untuk mengirim data secara *online*.
- Memeriksa sambungan Google Sheet, apakah Google Sheet sudah berfungsi untuk menyimpan data yang diunggah oleh mikrokontroler.

3.3 Perancangan Hardware

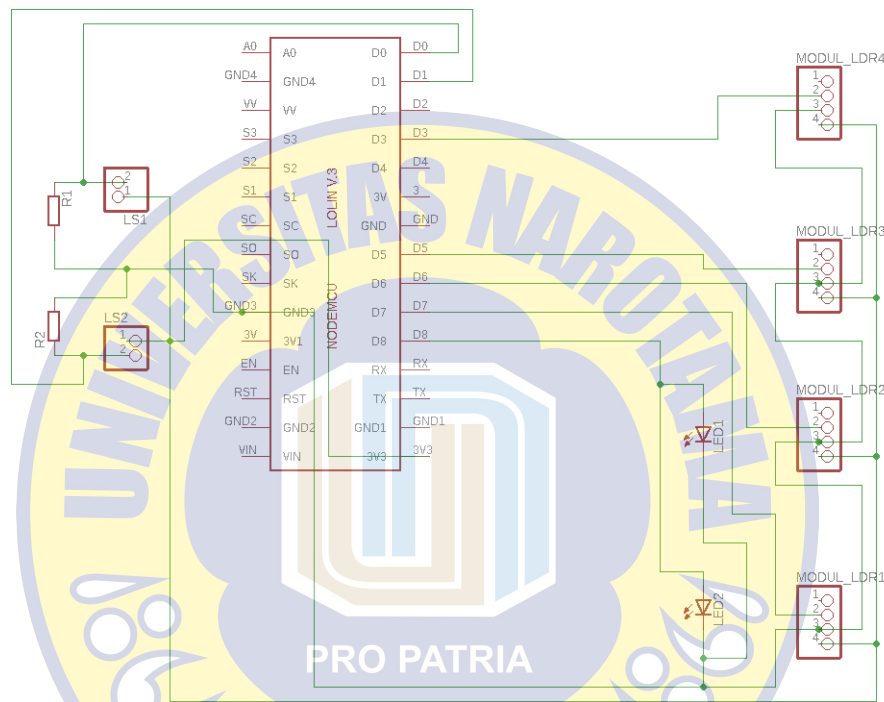
Setelah melakukan kajian-kajian dari penelitian terdahulu yang berkaitan dengan perancangan perangkat keras untuk Record Activity Sepeda Motor Berbasis IoT, berikut ini adalah kebutuhan hardware yang di perlukan:

1. Mikrokontroler, adalah sebuah chip yang pada penelitian ini digunakan untuk mengontrol rangkaian elektronik yang dapat

menyimpan program didalamnya. Chip pada mikrokontroller dapat mengontrol elektronik dengan sebuah program. Mikrocontroller di sini menggunakan NodeMCU ESP8266 yang juga memiliki modul wifi yang dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP. Dengan fungsi inilah nantinya semua data dari sensor-sensor yang diolah mikrokontroller diteruskan ke web server melalui koneksi TCP/IP. Sehingga data-data tersebut dapat diakses melalui web server yang dalam hal ini penulis menggunakan sarana Google Sheet sebagai web servernya.

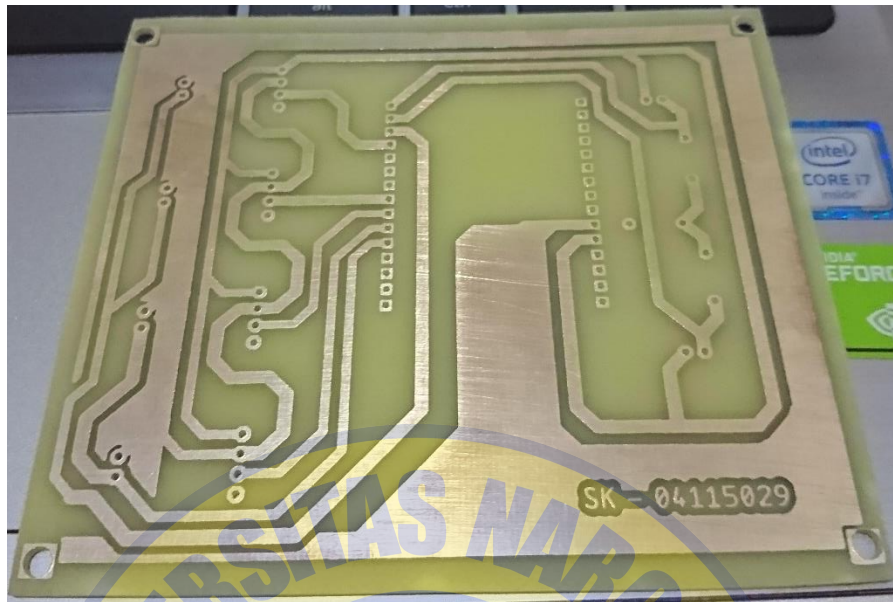
2. Sensor Cahaya, adalah sensor yang dalam hal ini menggunakan Module Sensor LDR berfungsi sebagai masukan untuk mendeteksi cahaya pada aktifitas lampu yang terdapat pada sepeda motor. Meliputi lampu sein, lampu depan serta lampu belakang. Sensor cahaya tersebut bekerja dengan memanfaatkan intensitas cahaya yang ada pada lampu-lampu tersebut, kemudian intensitas yang tertangkap diolah oleh mikrokontroller kemudian diunggah ke web server.
3. Sensor Limit Switch, berfungsi sebagai untuk mendeteksi masukan aktifitas dari pemakaian rem depan maupun rem belakang selama pemakaian sepeda motor. Sensor ini bekerja dengan memanfaatkan penekanan pada tuas rem depan maupun belakang sehingga dengan adanya penekanan tuas rem tersebut,

maka sensor akan menghasilkan nilai digit 1 sebagai indikator adanya aktifitas rem. Nilai 1 tersebut kemudian diolah oleh mikrokontroller dan diteruskan untuk diunggah dan ditampilkan ke web server.



Gambar 3.4 Desain Skema Hardware

Dalam perancangan rangkaian Record Activity Sepeda Motor Berbasis IoT, jika hasil pemantauan aktifitas kurang akurat atau tidak sesuai dengan harapan, maka akan diprogram ulang dan diuji ulang lagi sampai mendapat hasil yang sesuai dengan yang diharapkan. Kemudian selanjutnya dapat diuji secara keseluruhan sampai didapat hasil yang terbaik. Gambar 3.4 adalah perancangan *hardware* dari Record Activity Sepeda Motor Berbasis IoT berupa skematik yang meliputi Sensor – sensor dan Mikrokontroller yang dibuat menggunakan aplikasi perancang Eagle CadSoft.



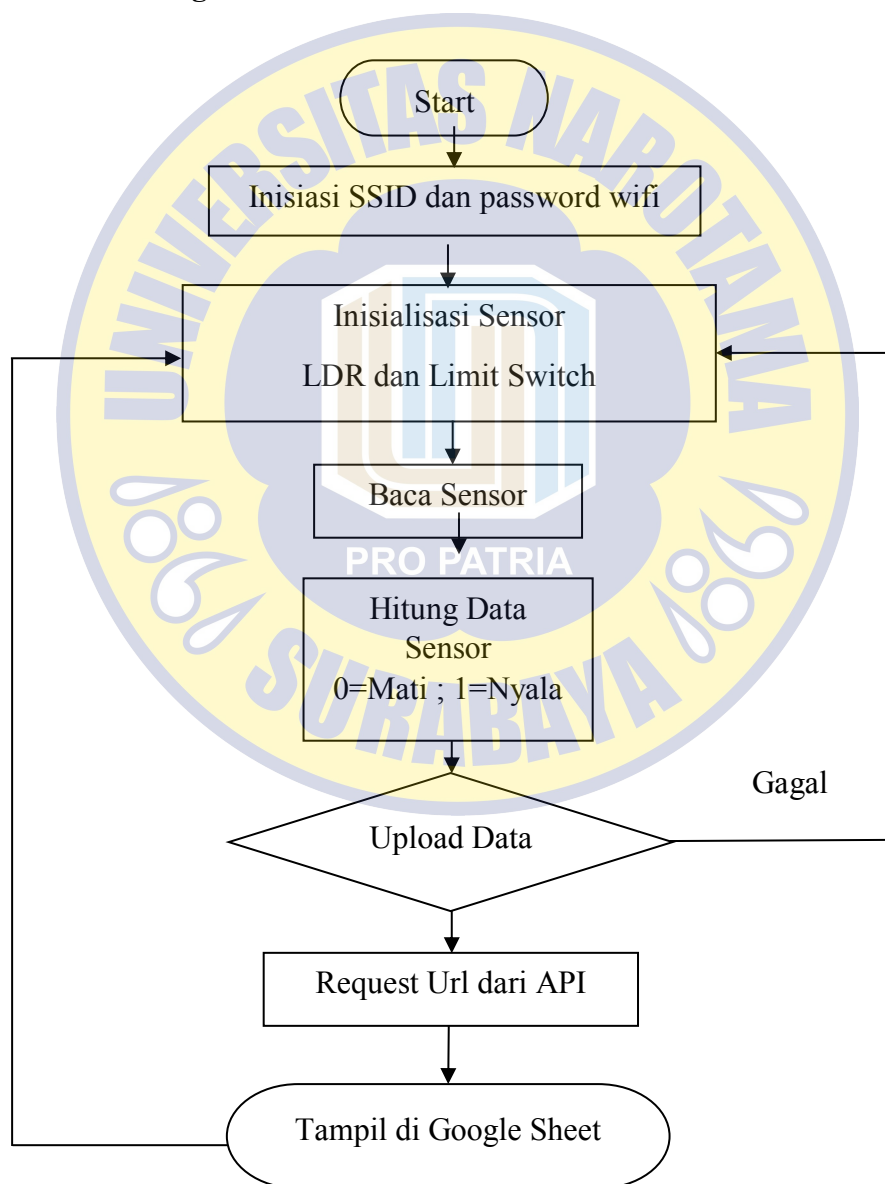
Gambar 3.5 PCB yang Sudah Dicetak

Tabel 3.1 Konfigurasi Pin

No	Komponen Pin	NodeMCU Pin
Sensor Module LDR1		
1	Vcc	3V
	Ground	Gnd
	DO	D7
Sensor Module LDR2		
2	Vcc	3V
	Ground	Gnd
	DO	D6
Sensor Module LDR3		
3	Vcc	3V
	Ground	Gnd
	DO	D5
Sensor Module LDR4		
4	Vcc	3V
	Ground	Gnd
	DO	D3
Limit Switch1		
5	NO	V
	C	D0
Limit Switch2		
6	NO	V
	C	D1
	Vcc	3V

Langkah pertama adalah dengan memasang seperti dalam tabel 3.1, sensor modul LDR dengan mikrokontroller menggunakan kabel *jumper*. Selanjutnya memasang Limit Switch dengan mikrokontroller, namun dalam hal ini Limit Switch menggunakan Resistor 220k dalam rangkaian skema.

3.4 Perancangan Software



Gambar 3.6 Flowchart Program Record Activity

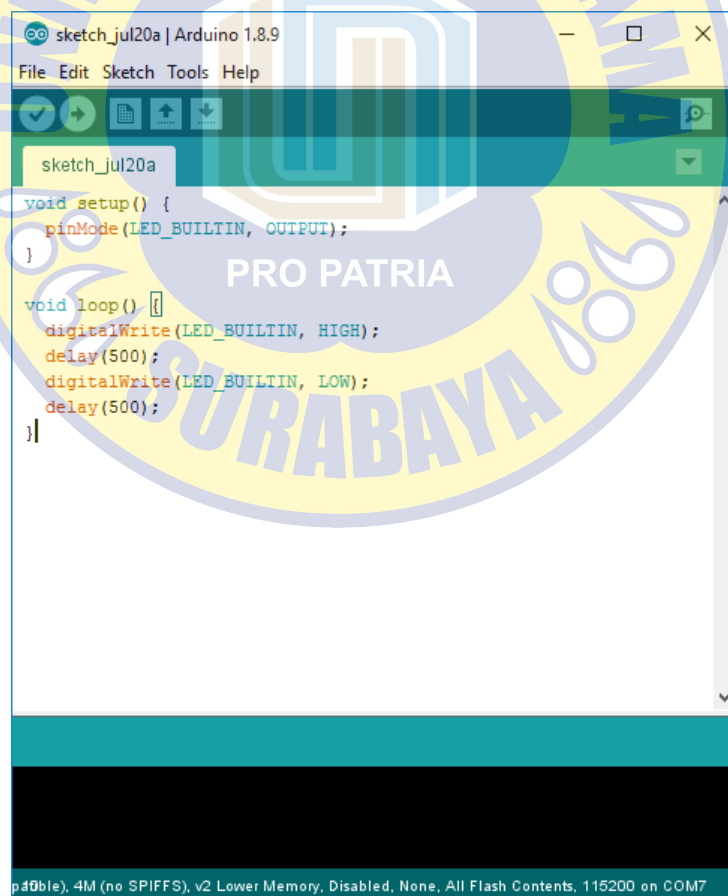
Perancangan perangkat lunak (*software*) meliputi perancangan system pendeteksi aktifitas rem dan lampu sepeda motor dengan pemrograman khusus NodeMCU ESP8266. Perancangan algoritma pendeteksi aktifitas dengan software arduino IDE, digunakan untuk mendeteksi aktifitas rem dan lampu dengan memanfaatkan masukan dari sensor LDR dan Limit Switch. Nilai yang dihasilkan akan diproses oleh algoritma yang telah dimasukkan dalam mikrokontroller untuk menghasilkan keluaran data berupa status deteksi aktifitas rem dan lampu. Setelah diproses oleh mikrokontroller, data-data tersebut akan diupload oleh NodeMCU ESP8266 agar dapat ditampilkan dan diakses melalui Google Sheet secara *online*. Agar mendapat hasil unggahan yang cepat dan akurat, diperlukan sambungan jaringan yang stabil dan kuat.

Pada gambar 3.6, menjelaskan alur sistem kerja perekam aktifitas sepeda motor berbasis IoT. Dimulai dari alat memeriksa sambungan jaringan, kemudian menerima hasil pembacaan sensor LDR dan limit switch untuk diolah mikrokontroller dan diunggah melalui sambungan internet, jika upload gagal, akan kembali untuk membaca sensor lagi. Namun jika sambungan berhasil maka mikrokontroller akan meminta URL dari API sehingga dapat tampil di kotak cell google sheet dan berulang lagi membaca sensor dari awal.

Program yang akan dibuat dengan menggunakan bahasa khusus NodeMCU ESP8266 yaitu bahasa yang mirip dengan bahasa pemrograman C++ dengan editor teksnya adalah arduino IDE. Pada *software* ini juga disediakan pustaka-pustaka yang dapat digunakan untuk berkomunikasi dengan perangkat seperti modul tambahan, sensor maupun mikrokontroller lainnya yang mendukung editor Arduino

IDE. Sehingga pemrograman dapat lebih mudah. Setelah proses memasukkan sumber kode pemrograman dalam Arduino IDE maka selanjutnya akan di-*compile* kemudian diunggah ke dalam Mikrokontroler NodeMCU ESP8266, file sumber kode yang akan dihasilkan adalah berekstensi .ino.

Dalam proses ini penulis melakukan instalasi awal dengan menginstal aplikasi arduino IDE yang dapat diunduh di web resmi arduino dengan nama web *arduino.cc* , tahap ini akan memproses pemrograman software yang akan ditanam pada mikrokontroler arduino, Berikut adalah contoh sumber kode untuk memeriksa antara arduino IDE dan NodeMCU ESP8266:



```
sketch_jul20a
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

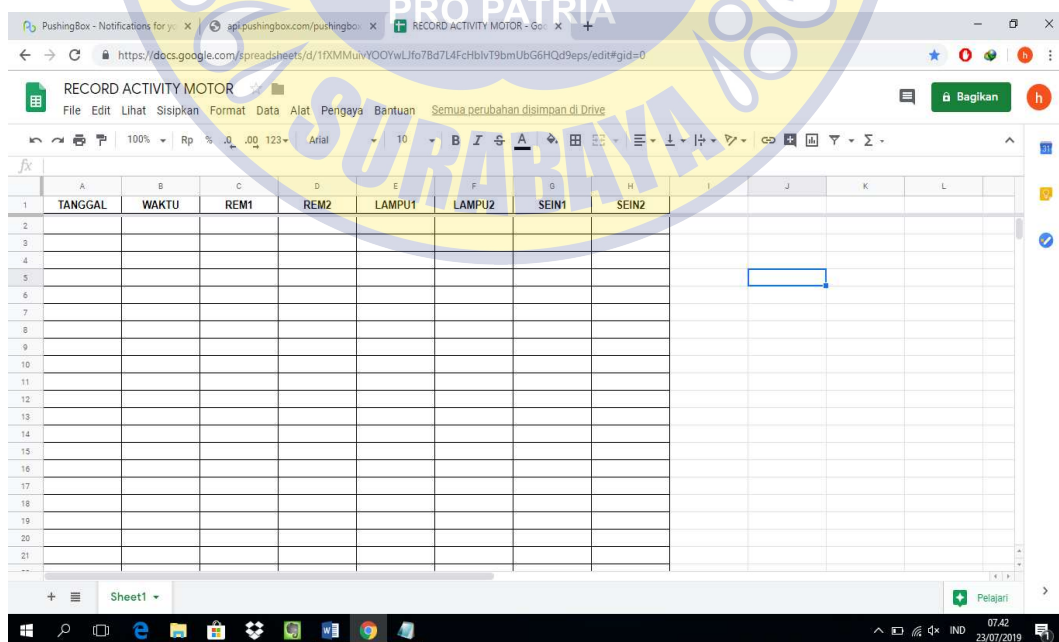
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
}
```

Gambar 3.7 Kode Pengecekan NodeMCU ESP8266

Pengujian antara arduino IDE dengan NodeMCU ESP8266 sangat diperlukan untuk memastikan bahwa perangkat NodeMCU ESP8266 benar-benar berfungsi dengan lancar. Kemudian *code* di atas di-*upload* ke NodeMCU ESP8266. Arduino dapat dipastikan normal jika led yang ada di papan Arduino berjalan sesuai dengan keinginan.

3.4.1 Perancangan Tampilan Google Sheet

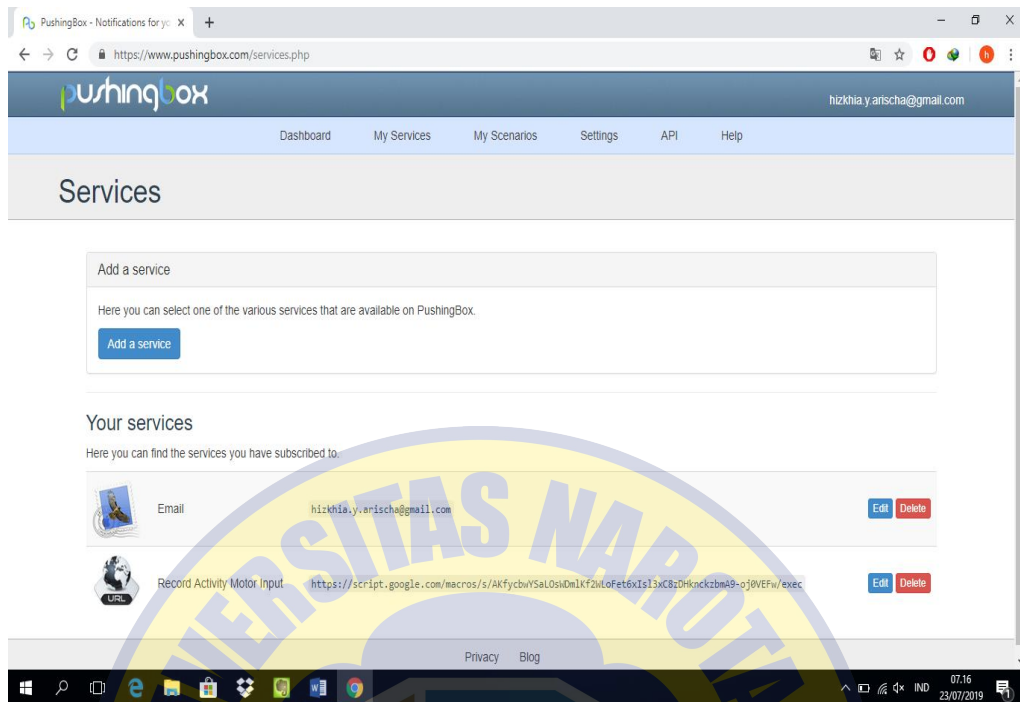
Tampilan hasil akhir dalam penelitian ini diharapkan dalam bentuk tampilan tabel yang dapat diunggah secara *realtime* dan diakses secara online. Sebelum mendapatkan hasil tersebut, diperlukan adanya perancangan tampilan hasil akhirnya. Dalam hal ini, peneliti menggunakan fasilitas Google Sheet sebagai tampilan hasil akhirnya.



Gambar 3.8 Tampilan Output Awal Google Sheet

Agar data keluaran dapat tampil dalam Google Sheet, langkah pertama adalah mengkode NodeMCU ESP8266 agar terhubung dengan sambungan *wifi* dan mengatur interval waktu unggahan data, misal interval per menit atau per detik. Selanjutnya simpan koding tersebut dan berlanjut untuk membuat lembar Google Sheet. Dari bilah baris alamat URL Google Sheet, salin string antara `d / dan / edit` dan simpan di suatu tempat. Ini adalah kunci berbagi unik spreadsheet yang nantinya akan dibutuhkan dalam skrip Google Apps Script. Untuk membuat skrip Google Apps, pergi ke menu Alat pilih Editor Skrip dari Google Sheets. Di bagian kode tersebut, diisi dengan kode yang berisi fungsi untuk menampilkan kolom-kolom data yang diunggah oleh NodeMCU ESP 8266 serta kunci berbagi spreadsheet yang telah disalin dari URL spreadsheet sebelumnya. Langkah selanjutnya adalah mempublikasikan skrip Google Apps sehingga dapat diakses melalui URL. Untuk melakukannya, pilih menu “Publikasikan” pada baris *toolbar* pada lembar Google Apps Script kemudian klik sub menu “terapkan sebagai Aplikasi Web”. Selama penerbitan, Google akan meminta untuk memberikan beberapa izin akses file Google Sheet yang telah dibuat.

Diperlukan login di platform Pushing Box menggunakan akun gmail untuk membuat services dengan pilihan custom URL. Di dalam custom URL, berguna untuk membuat service yang nantinya untuk menjembatani antara data yang dikirim mikrokontroler agar dapat tampil di lembar Google Sheet dengan memasukkan URL dari Google Apps Script yang telah dipublikasikan dalam tahap sebelumnya. Gambar 3.9 adalah tampilan services di pushingbox beserta URL dari Google Apps Script.



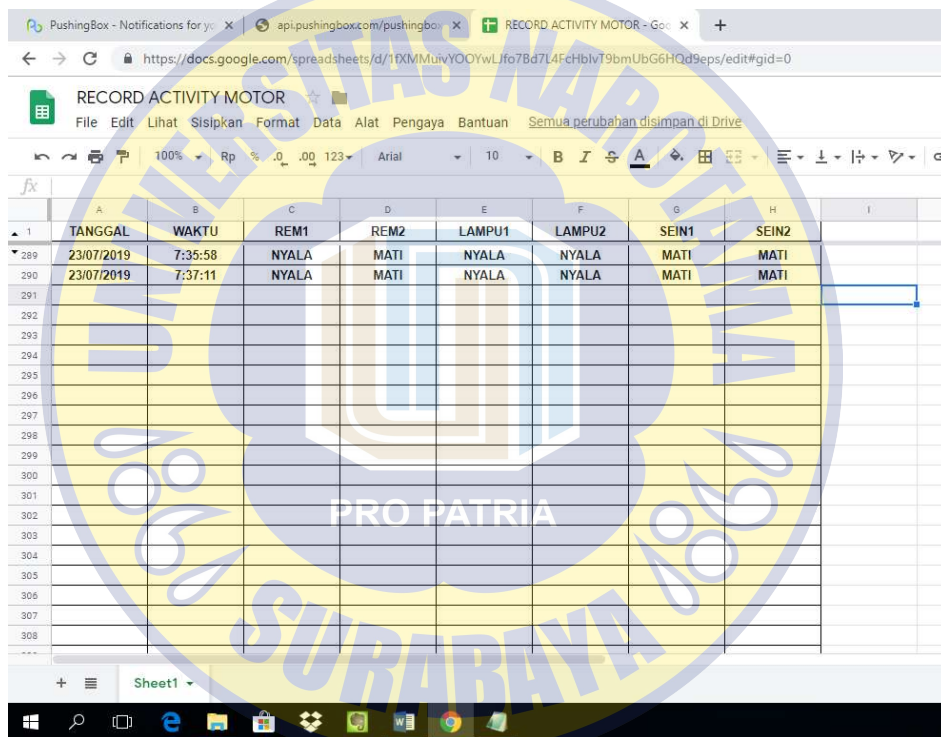
Gambar 3.9 Service pushingbox



Gambar 3.10 Skenario pushingbox

Pada gambar 3.10, mengatur *request* URL yang berisi isian *variable sheet* di kolom Google Sheet yang isinya merupakan hasil data pembacaan sensor yang telah

diolah dalam mikrokontroler yang telah dikonversikan ke dalam kondisi “nyala” dan “mati”. Dalam hal ini skenario data yang akan diupload ke google melalui *service request* URL pushingbox. Langkah terakhir adalah memeriksa koneksi antara pushingbox dengan google sheet yaitu dengan menyalin URL aplikasi web yang didapat dari Pushing Box beserta isian data variable sheet kemudian tempel string tersebut di bidang URL browser web.



Gambar 3.11 Tampilan Ujicoba Google Sheet