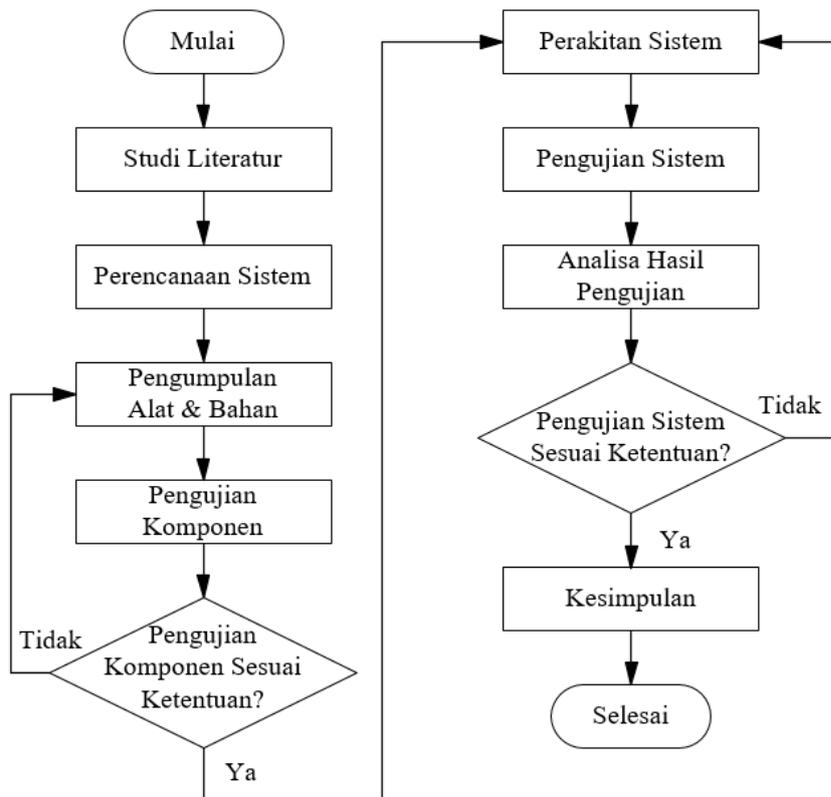


BAB III
METODE PENELITIAN

3.1 Flowchart Penelitian



Gambar 3.1. *Flowchart* Penelitian

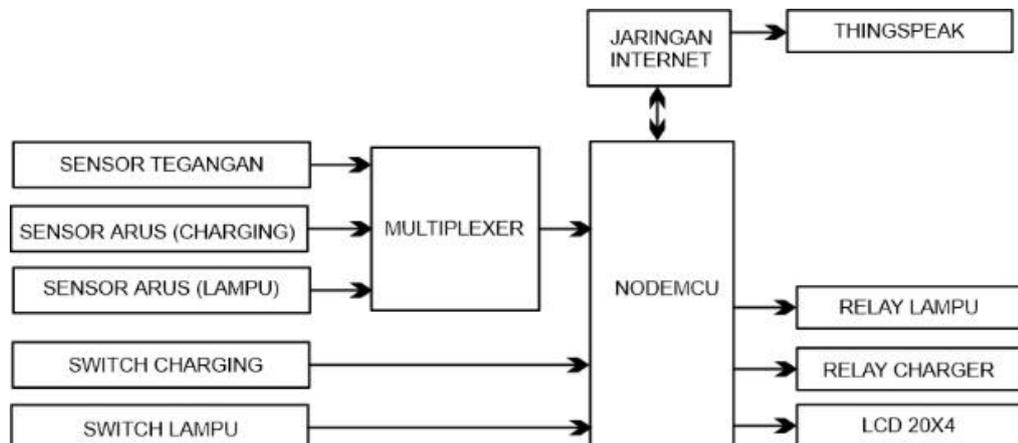
Gambar 3.1 menunjukkan *flowchart* dari penelitian pada tugas akhir ini.

Proses pada setiap tahapan dijelaskan seperti berikut ini:

3.1.1 Perencanaan Sistem

Perencanaan sistem dilakukan dengan merancang sistem kerja alat penelitian. Tahap ini membantu mengetahui komponen-komponen yang dibutuhkan dan juga dapat mengatur tata letak sambungan dari komponen satu

ke komponen lainnya guna menciptakan alat dengan sistem kerja yang diharapkan.



Gambar 3.2. Arsitektur Sistem

Gambar 3.2 menunjukkan arsitektur sistem dari kerja alat yang dibangun pada penelitian ini dan bertujuan untuk mempermudah dalam mendeskripsikan kerja dari keseluruhan sistem yang dibuat. Berikut alur penjelasan dari arsitektur sistem pada Gambar 3.2:

- Sensor tegangan, sensor arus (*charging*), dan sensor arus (lampu/*discharging*) dihubungkan dengan *multiplexer* sebagai penyedia pin *ADC* tambahan untuk perangkat yang akan terhubung dengan NodeMCU. Sensor tegangan dan sensor arus ini bertugas sebagai *input* data bagi NodeMCU.
- *Multiplexer* dihubungkan dengan NodeMCU agar komponen yang terhubung dengan *multiplexer* dapat terhubung dengan NodeMCU.
- *Switch charging* dan *switch* lampu berperan sebagai pemberi perintah pada NodeMCU untuk memutus atau menghubungkan proses *charging* maupun proses *discharging* yang kemudian perintah tersebut akan diteruskan oleh

NodeMCU ke relay lampu dan relay *charger*. Kedua buah relay tersebut akan diatur sedemikian rupa sesuai program yang telah di input pada NodeMCU agar proses *charging* dan proses *discharging* tidak berlangsung secara bersamaan.

- Data-data dari sensor-sensor yang telah diterima NodeMCU akan diolah dan ditampilkan pada LCD 20×4 sebagai penampil hasil monitoring untuk penggunaan luring / *offline*.
- NodeMCU akan terhubung dengan jaringan internet agar sistem monitoring dapat diakses pada *ThinkSpeak* sebagai fasilitas monitoring secara daring / *online*.

3.1.2 Pengumpulan Alat Dan Bahan

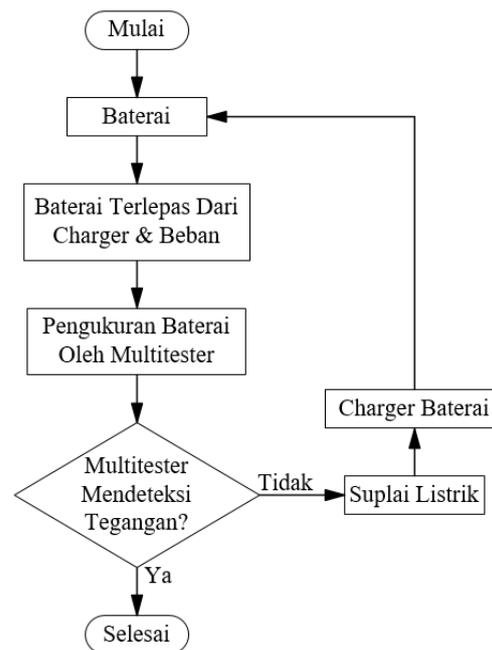
Pada tahap ini merupakan proses kelanjutan dari perencanaan sistem dimana pengumpulan alat dan bahan dilakukan guna mengefisienkan waktu dalam membangun sistem yang diharapkan pada penelitian. Komponen-komponen yang dibutuhkan diantaranya adalah baterai Li-Ion 3,7V, *multiplexer*, NodeMCU, kabel jumper, LCD 16×2 I2C, saklar spat, saklar MCB, sensor arus ACS712, dan sensor tegangan DC 0 – 25V.

3.1.3 Pengujian Komponen

Pengujian komponen dilakukan untuk mengetahui kualitas dari setiap komponen yang digunakan. Semakin kecil nilai *error* komponen mengindikasikan semakin handalnya komponen tersebut. Jika hasil pengujian komponen buruk (komponen rusak/mati/nilai *error* < 5%) maka dilakukan pengumpulan alat dan bahan ulang untuk komponen-komponen bermasalah

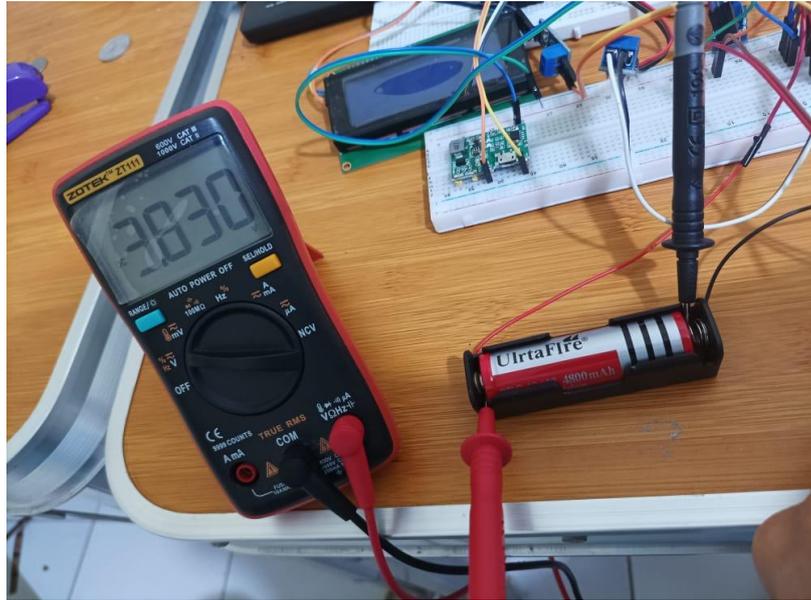
sampai dengan seluruh komponen yang digunakan memiliki kualitas yang bagus (dapat menyala dan nilai *error* > 5%). Komponen-komponen yang dilakukan pengujian adalah sebagai berikut:

3.1.3.1 Pengujian Baterai Li-Ion 3,7V



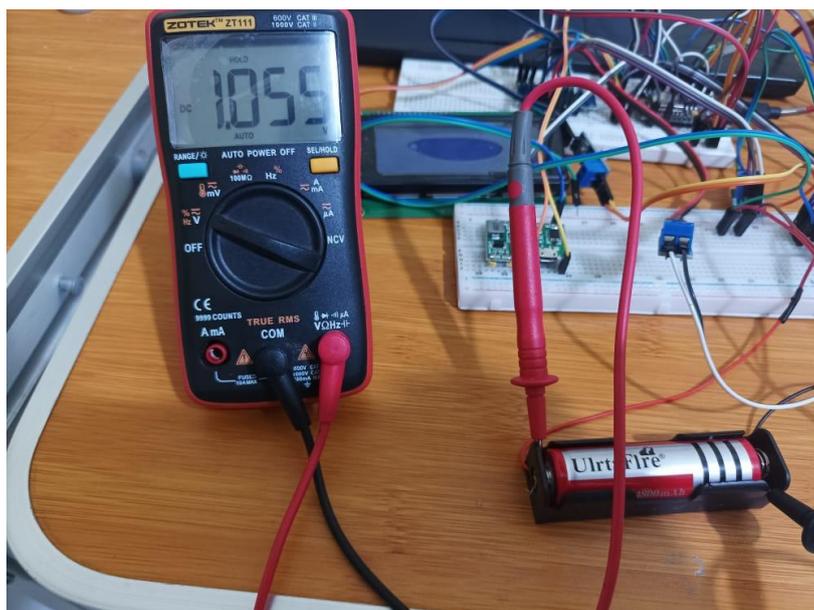
Gambar 3.3. *Flowchart* Pengujian Baterai Li-Ion 3,7V

Gambar 3.3. menunjukkan *flowchart* dari pengujian baterai Li-Ion 3,7V dimana pengujian dilakukan ketika baterai tidak tersambung apapun kecuali alat ukur (terputus dengan suplai listrik dan beban). Multimeter akan mengukur nilai tegangan dari baterai. Jika tidak ada tegangan yang terukur dari baterai maka dilakukan *charging* terlebih dahulu. Dan setelah pengecasan selesai, baterai harus terlepas kembali dari *charge* baterai dan dilakukan pengukuran ulang sampai didapatkan hasil pengukuran dari nilai tegangan baterai.



Gambar 3.4. Pengukuran Tegangan Maksimum Baterai

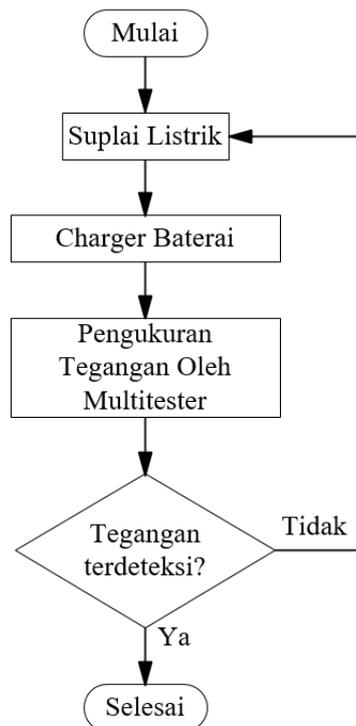
Gambar 3.4 menunjukkan pengukuran tegangan maksimum baterai yang bertujuan untuk mengetahui tegangan tertinggi dari baterai *lithium-ion* yang digunakan. Pengukuran tegangan maksimum baterai pada awal penggunaan baterai dapat menjadikan tolak ukur kedepannya apakah kondisi baterai masih bagus atau kondisi baterai mengalami penurunan.



Gambar 3.5. Pengukuran Tegangan Minimum Baterai

Gambar 3.5 menunjukkan pengukuran tegangan minimum baterai guna mengetahui tegangan terendah baterai yang dapat menyuplai beban. Pengujian tegangan baterai dilakukan dengan menghubungkan kutup positif baterai dengan *probe* V multimeter dan kutub negatif baterai dihubungkan dengan *probe* COM multimeter.

3.1.3.2 Pengujian *Charger* Baterai



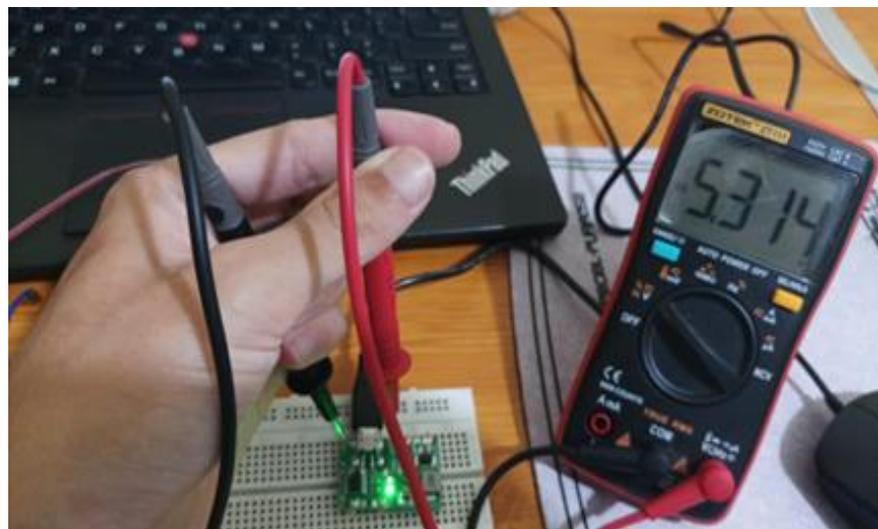
Gambar 3.6. *Flowchart* Pengujian *Charger* Baterai

Gambar 3.6 menunjukkan *flowchart* untuk pengujian *charger* baterai dimana *charger* baterai yang tersambung dengan suplai listrik akan diukur nilai tegangannya oleh multimeter. Apabila multimeter tidak mendeteksi nilai tegangan maka akan diulang dari proses penyambungan suplai listrik dengan *charger* baterai dan dilakukan pengukuran kembali sampai multimeter mendeteksi nilai tegangan *charger* baterai.



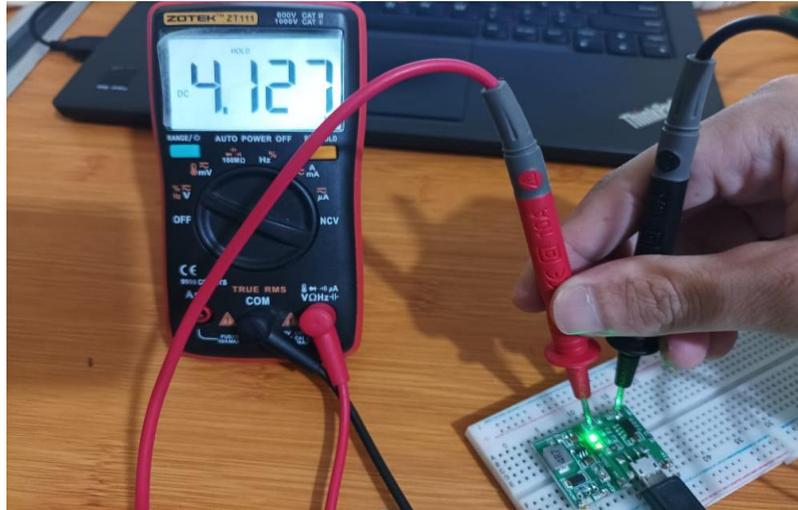
Gambar 3.7. Pengukuran Tegangan Sumber *Charger*

Gambar 3.7 menunjukkan pengukuran tegangan sumber *charger*, dimana pada penelitian ini sumber dari *charger* merupakan listrik PLN. Sehingga sumber listrik yang berasal dari PLN merupakan *input* untuk *charger*.



Gambar 3.8. Pengukuran Tegangan *Output Charger*

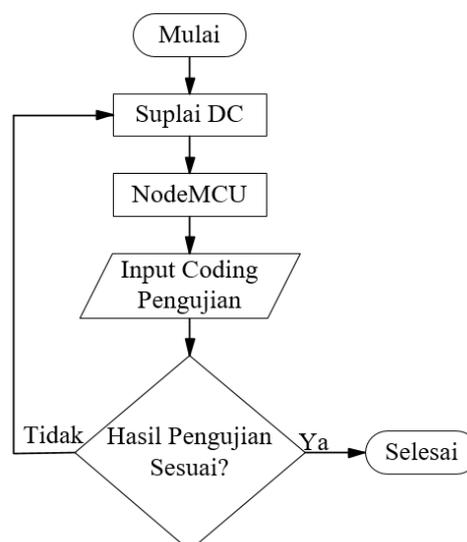
Gambar 3.8 menunjukkan pengukuran tegangan *output charger*. Pengujian ini bertujuan untuk mengetahui kesesuaian tegangan keluaran *charger* untuk nantinya disuplai ke beban *charging* (baterai). Pengukuran tegangan dilakukan menggunakan multimeter.



Gambar 3.9. Pengukuran Tegangan *Output* Modul *Charger*

Gambar 3.9. menunjukkan pengukuran tegangan *output* modul *charger* yang bertujuan untuk mengetahui besar keluaran dari proses akhir perubahan listrik AC dari PLN menjadi listrik DC yang sesuai dengan kebutuhan jenis tegangan baterai. Tegangan *input* baterai yang terlalu besar beresiko membuat baterai cepat rusak.

3.1.3.3 Pengujian NodeMCU



Gambar 3.10. *Flowchart* Pengujian NodeMCU

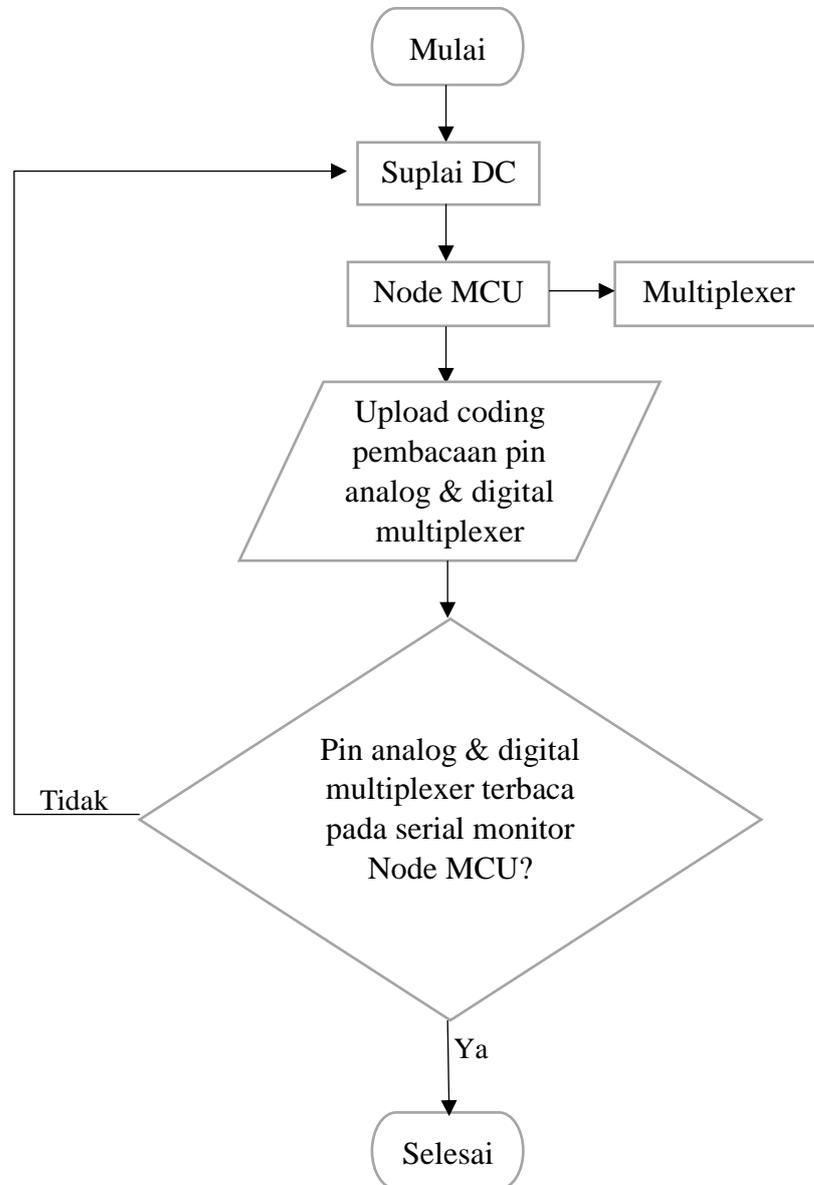
Gambar 3.10 menunjukkan *flowchart* yang digunakan untuk pengujian NodeMCU. Listrik yang dibutuhkan oleh NodeMCU adalah jenis listrik DC. Oleh sebab itu, NodeMCU harus disuplai oleh suplai DC. Untuk mengetahui apakah NodeMCU dapat beroperasi, maka diinputkan *coding* untuk pengujian. Jika hasil pengujian sesuai, maka pengujian selesai. Dan jika hasil pengujian tidak sesuai maka proses akan diulang dari penghubungan suplai DC hingga *input coding* pengujian sampai didapatkan hasil pengujian yang sesuai.

```
PENGUJIAN_NODEMCU | Arduino 1.8.15
File Edit Sketch Tools Help
Verify
PENGUJIAN_NODEMCU $
1 void setup()
2 {
3   Serial.begin(9600);
4 }
5
6 void loop()
7 {
8   Serial.println("ARFAN LESMANA - PENGUJIAN NODEMCU");
9   delay (5000);
10 }
```

Gambar 3.11. *Coding* Pengujian NodeMCU

Gambar 3.11 menunjukkan *coding* yang diupload pada NodeMCU dengan tujuan untuk mengetahui berfungsi atau tidaknya NodeMCU yang digunakan. NodeMCU ini merupakan *master* dari alat yang dibangun pada penelitian ini.

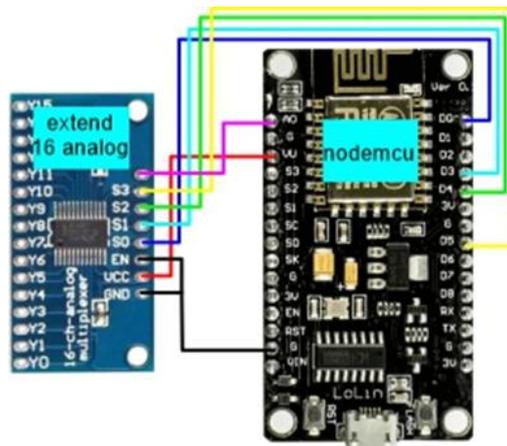
3.1.3.4 Pengujian *Multiplexer*



Gambar 3.12. *Flowchart* Pengujian *Multiplexer*

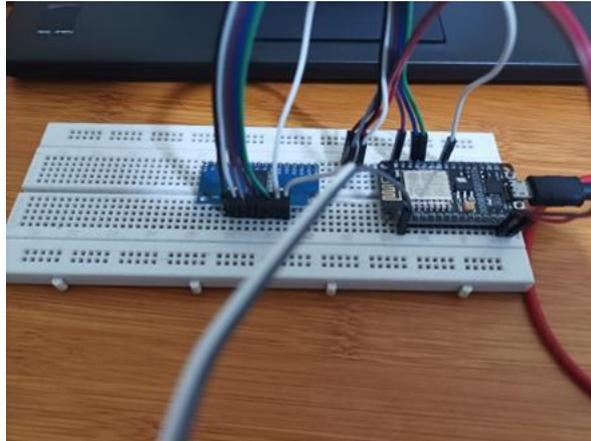
Gambar 3.12. menunjukkan *flowchart* pengujian *multiplexer* dimana pengujian ini berfungsi untuk mengetahui apakah *multiplexer* tersambung dengan NodeMCU sebagai master dan untuk mengetahui apakah pin-pin pada *multiplexer* dapat berfungsi sebagai *port* tambahan bagi NodeMCU sesuai dengan yang dibutuhkan. Proses pengujian ini dilakukan dengan

menghubungkan *power supply* dengan NodeMCU agar NodeMCU tersuplai listrik dan dapat beroperasi. Kemudian *multiplexer* dihubungkan dengan NodeMCU. Lalu NodeMCU di *input* data gerbang logika untuk mengoperasikan *multiplexer* melalui *coding* yang akan diupload pada NodeMCU. Jika hasil pengujian sesuai maka pengujian *multiplexer* selesai. Dan jika hasil pengujian tidak sesuai, maka proses diulang dimulai dari penyambungan *power supply* dengan NodeMCU sampai dengan mendapatkan hasil pengujian yang sesuai.



Gambar 3.13. *Wiring* Pengujian *Multiplexer*

Gambar 3.13 menunjukkan *wiring* pengujian *multiplexer* dimana pada pengujian ini *multiplexer* dihubungkan dengan NodeMCU dengan tujuan sebagai penyedia pin analog maupun digital untuk NodeMCU. Pin VCC dan GND *multiplexer* dihubungkan dengan pin VU dan G NodeMCU sebagai jalur daya untuk *multiplexer* yang disuplai oleh NodeMCU.



Gambar 3.14. Pengujian *Multiplexer*

Gambar 3.14 menunjukkan pengujian *multiplexer* dengan kesesuaian rangkaian pengujian yang ditunjukkan oleh Gambar 3.12. Pada pengujian ini, suplai listrik untuk *multiplexer* adalah NodeMCU dan suplai listrik untuk NodeMCU adalah laptop. Berikut ini merupakan *coding* yang diupload untuk pengujian *multiplexer*:

Tabel 3.1. *Coding* Pengujian NodeMCU

<i>Coding</i>	Keterangan
<pre>#define S0 D0 #define S1 D3 #define S2 D4 #define S3 D5 #define SIG A0</pre>	Bagian ini berfungsi sebagai pendeklarasi nomor pin pada <i>multiplexer</i> di NodeMCU
<pre>int decimal = 2; int adc_c0; int adc_c1; int adc_c2; int adc_c3; int adc_c4; int adc_c5; int adc_c6; int adc_c7; int adc_c8; int adc_c9; int adc_c10; int adc_c11; int adc_c12;</pre>	Pada bagian ini merupakan penggunaan variable <i>integer</i> untuk mengetahui beroperasi atau tidaknya masing-masing pin data pada <i>multiplexer</i> dengan menempatkan variabel “decimal” adalah 2

Coding	Keterangan
<pre>int adc_c13; int adc_c14; int adc_c15;</pre>	
<pre>void setup() { pinMode(S0, OUTPUT); pinMode(S1, OUTPUT); pinMode(S2, OUTPUT); pinMode(S3, OUTPUT); pinMode(SIG, INPUT); Serial.begin(9600); }</pre>	<p>Bagian ini merupakan bentuk inisialisasi atau pengenalan untuk pendefinisian mode pin <i>multiplexer</i> pada NodeMCU.</p>
<pre>void loop() { digitalWrite(S0, LOW); digitalWrite(S1, LOW); digitalWrite(S2, LOW); digitalWrite(S3, LOW); adc_c0 = analogRead(SIG); digitalWrite(S0, HIGH); digitalWrite(S1, LOW); digitalWrite(S2, LOW); digitalWrite(S3, LOW); adc_c1 = analogRead(SIG); digitalWrite(S0, LOW); digitalWrite(S1, HIGH); digitalWrite(S2, LOW); digitalWrite(S3, LOW); adc_c2 = analogRead(SIG); digitalWrite(S0, HIGH); digitalWrite(S1, HIGH); digitalWrite(S2, LOW); digitalWrite(S3, LOW); adc_c3 = analogRead(SIG); digitalWrite(S0, LOW); digitalWrite(S1, LOW); digitalWrite(S2, HIGH); digitalWrite(S3, LOW); adc_c4 = analogRead(SIG); digitalWrite(S0, HIGH); digitalWrite(S1, LOW); digitalWrite(S2, HIGH); digitalWrite(S3, LOW); adc_c5 = analogRead(SIG); }</pre>	<p>Pada bagian ini, program akan dijalankan secara terus-menerus terhadap pin-pin pada <i>multiplexer</i> oleh NodeMCU secara berurutan hingga program berhenti di jalankan.</p>

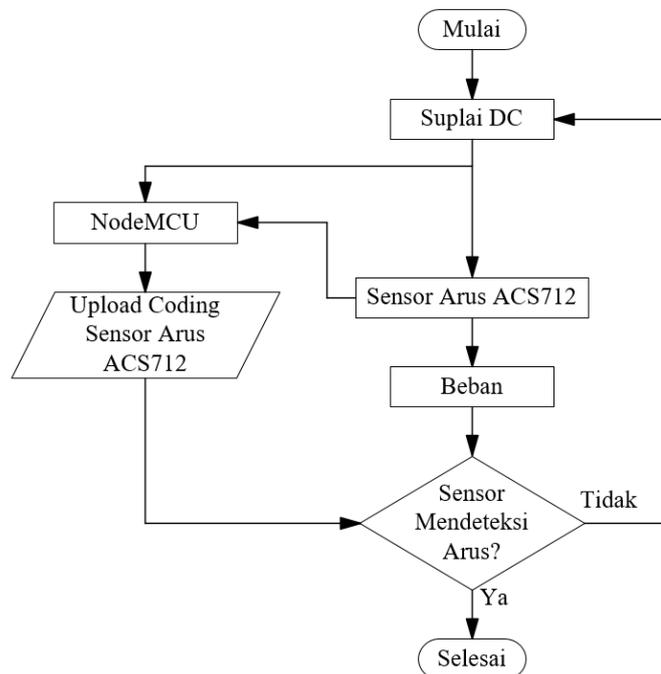
<i>Coding</i>	Keterangan
<pre>digitalWrite(S0, LOW); digitalWrite(S1, HIGH); digitalWrite(S2, HIGH); digitalWrite(S3, LOW); adc_c6 = analogRead(SIG); digitalWrite(S0, HIGH); digitalWrite(S1, HIGH); digitalWrite(S2, HIGH); digitalWrite(S3, LOW); adc_c7 = analogRead(SIG); digitalWrite(S0, LOW); digitalWrite(S1, LOW); digitalWrite(S2, LOW); digitalWrite(S3, HIGH); adc_c8 = analogRead(SIG); digitalWrite(S0, HIGH); digitalWrite(S1, LOW); digitalWrite(S2, LOW); digitalWrite(S3, HIGH); adc_c9 = analogRead(SIG); digitalWrite(S0, LOW); digitalWrite(S1, HIGH); digitalWrite(S2, LOW); digitalWrite(S3, HIGH); adc_c10 = analogRead(SIG); digitalWrite(S0, HIGH); digitalWrite(S1, HIGH); digitalWrite(S2, LOW); digitalWrite(S3, HIGH); adc_c11 = analogRead(SIG); digitalWrite(S0, LOW); digitalWrite(S1, LOW); digitalWrite(S2, HIGH); digitalWrite(S3, HIGH); adc_c12 = analogRead(SIG); digitalWrite(S0, HIGH); digitalWrite(S1, LOW); digitalWrite(S2, HIGH); digitalWrite(S3, HIGH); adc_c13 = analogRead(SIG); digitalWrite(S0, LOW); digitalWrite(S1, HIGH); digitalWrite(S2, HIGH); digitalWrite(S3, HIGH); adc_c14 = analogRead(SIG);</pre>	

Coding	Keterangan
<pre>digitalWrite(S0, HIGH); digitalWrite(S1, HIGH); digitalWrite(S2, HIGH); digitalWrite(S3, HIGH); adc_c15 = analogRead(SIG);</pre>	
<pre>Serial.print("adc c0: "); Serial.println(adc_c0); Serial.print("adc c1: "); Serial.println(adc_c1); Serial.print("adc c2: "); Serial.println(adc_c2); Serial.print("adc c3: "); Serial.println(adc_c3); Serial.print("adc c4: "); Serial.println(adc_c4); Serial.print("adc c5: "); Serial.println(adc_c5); Serial.print("adc c6: "); Serial.println(adc_c6); Serial.print("adc c7: "); Serial.println(adc_c7); Serial.print("adc c8: "); Serial.println(adc_c8); Serial.print("adc c9: "); Serial.println(adc_c9); Serial.print("adc c10: "); Serial.println(adc_c10); Serial.print("adc c11: "); Serial.println(adc_c11); Serial.print("adc c12: "); Serial.println(adc_c12); Serial.print("adc c13: "); Serial.println(adc_c13); Serial.print("adc c14: "); Serial.println(adc_c14); Serial.print("adc c15: "); Serial.println(adc_c15); Serial.println(); delay(1000); }</pre>	<p>Bagian ini berfungsi untuk mengirimkan data ke <i>serial port multiplexer</i> dan NodeMCU dan menampilkannya di serial monitor pada aplikasi Arduino.</p>

Berdasarkan Tabel 3.1, data pada *coding* tersebut diinput ke NodeMCU yang telah terhubung dengan *multiplexer* yang dirangkai sesuai

Gambar 3.13 untuk mengetahui terhubung, bekerja, dan sinkron atau tidaknya *multiplexer* dengan NodeMCU. *Coding* tersebut diupload pada aplikasi Arduino yang terinstal pada sebuah perangkat dan perangkat tersebut terhubung dengan NodeMCU.

3.1.3.5 Pengujian Sensor Arus ACS712



Gambar 3.15. *Flowchart* Pengujian Sensor Arus ACS712

Gambar 3.15. menunjukkan *flowchart* dari pengujian sensor arus ACS712 dimulai dengan menghubungkan suplai listrik dengan NodeMCU. Sensor arus ACS712 diletakkan di antara suplai listrik dengan beban dan dihubungkan dengan NodeMCU. Setelah sensor terhubung dengan NodeMCU, *uploading coding* sensor arus ACS712 dilakukan agar sensor dapat berjalan. Jika sensor mendeteksi nilai arus yang mengalir antara sumber (suplai listrik) menuju beban, pengujian selesai. Jika sensor

tidak mendeteksi nilai arus, maka dilakukan proses pengujian ulang dimulai dari awal.

Pengukuran yang dilakukan oleh sensor memiliki nilai maksimum $\%error$, yaitu sebesar 5%. $Error$ pengukuran pada sensor-sensor yang ada dalam sistem dan alat ukur dapat dihitung melalui persamaan (3.1) (Leny & Haryudo, 2019):

$$\%Error = \left| \frac{(\text{Nilai Sebenarnya} - \text{Nilai Terbaca})}{\text{Nilai Sebenarnya}} \right| \times 100\% \quad (3.1)$$

Data pada persamaan 3.1 terdiri dari $\%error$, merupakan persentase kesalahan pengukuran sensor. Nilai sebenarnya merupakan nilai hasil pengukuran dari alat ukur yang telah dikalibrasi dan sesuai dengan standar internasional (contohnya adalah seperti multimeter). Nilai terbaca merupakan nilai hasil pengukuran sensor. Nilai sebenarnya merupakan nilai pembanding untuk nilai terbaca. Dalam hal ini, multimeter akan menjadi pembanding sensor dalam melakukan pengukuran.

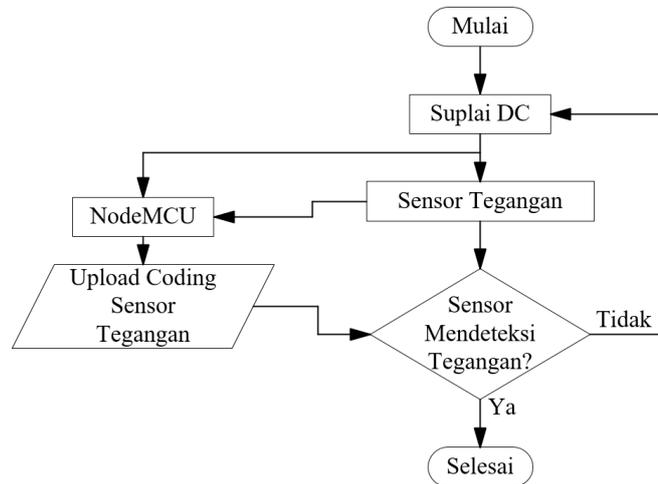
Setelah mengetahui nilai $\%error$ dari sensor, maka dapat diketahui persentase akurasi sensor tersebut dengan menggunakan persamaan (3.2):

$$\%Akurasi = 100\% - \%Error \quad (3.2)$$

Persentase akurasi (% akurasi) merupakan ketepatan sensor dalam melakukan pengukuran. Sedangkan, untuk mengetahui rata-rata dari keseluruhan hasil pengukuran, perhitungan dilakukan menggunakan persamaan (3.3):

$$Rata - rata = \frac{Jumlah\ Hasil\ Pengukuran}{Banyaknya\ Pengukuran\ Yang\ Dilakukan} \quad (3.3)$$

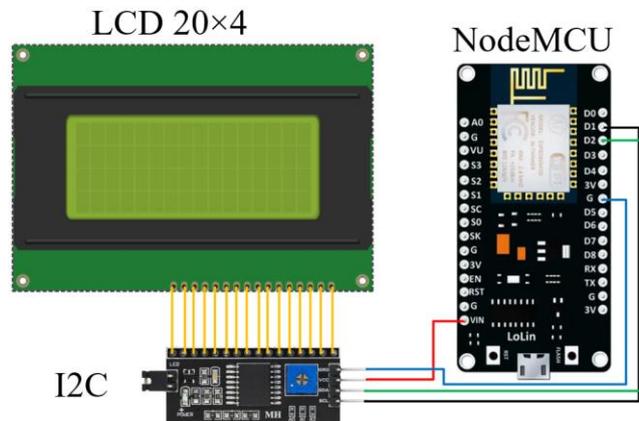
3.1.3.6 Pengujian Sensor Tegangan DC 0 – 25V



Gambar 3.16. *Flowchart* Pengujian Sensor Tegangan

Gambar 3.16 menunjukkan *flowchart* dari pengujian sensor tegangan DC 0 – 25V. Suplai listrik terhubung dengan NodeMCU. Sensor diberikan suplai listrik agar dapat menyala. *Output* sensor tegangan dihubungkan ke NodeMCU dan *coding* diupload melalui NodeMCU. Jika hasil pengujian memberikan kondisi dimana sensor dapat mendeteksi, maka pengujian selesai. Dan jika sebaliknya, maka proses pengujian diulang dari awal.

3.1.3.7 Pengujian LCD I2C 20×4



Gambar 3.17. *Wiring* Pengujian LCD I2C 20×4

Gambar 3.17 menunjukkan *wiring* dari LCD 20×4 dengan I2C dan dari I2C menuju nodeMCU. Pin SCL dan SDA I2C yang terhubung dengan pin D1 dan D2 nodeMCU berfungsi untuk mengirimkan data dari nodeMCU untuk ditampilkan pada LCD 20×4. Pin *ground* dan VCC I2C dihubungkan dengan pin *ground* dan VIN nodeMCU sebagai suplai listrik untuk menyalakan LCD 20×4.

```
PENGUJIAN_LCDI2C20X4
1 #include<Wire.h>
2 #include<LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x27, 20, 4);
4
5 void setup()
6 {
7   lcd.begin();
8   lcd.backlight();
9   lcd.setCursor(2, 0);
10  lcd.print("PENGUJIAN LCD_I2C");
11  lcd.setCursor(3, 1);
12  lcd.print("ARFAN LESMANA");
13  lcd.setCursor(5, 2);
14  lcd.print("167002017");
15  lcd.setCursor(2, 3);
16  lcd.print("TEKNIK ELEKTRO");
17  delay(1500);
18  lcd.clear();
19 }
20
21 void loop()
22 {
23   lcd.setCursor(7, 1);
24   lcd.print("DONE");
25   delay(1000);
26 }
```

Gambar 3.18. *Coding* LCD I2C 20×4

Gambar 3.18 merupakan *coding* yang diupload pada nodeMCU yang terhubung dengan LCD I2C 20×4 sesuai dengan *wiring* pada Gambar 3.17. Pada *coding* ini diinputkan perintah mengenai apa saja yang akan ditampilkan pada LCD.

3.1.3.8 Pengujian Saklar

Saklar yang digunakan pada penelitian ini adalah tiga buah saklar *Single Pole Single Throw* (SPST). Pengujian dilakukan untuk mengetahui kesesuaian jenis saklar yang digunakan dengan fungsinya.

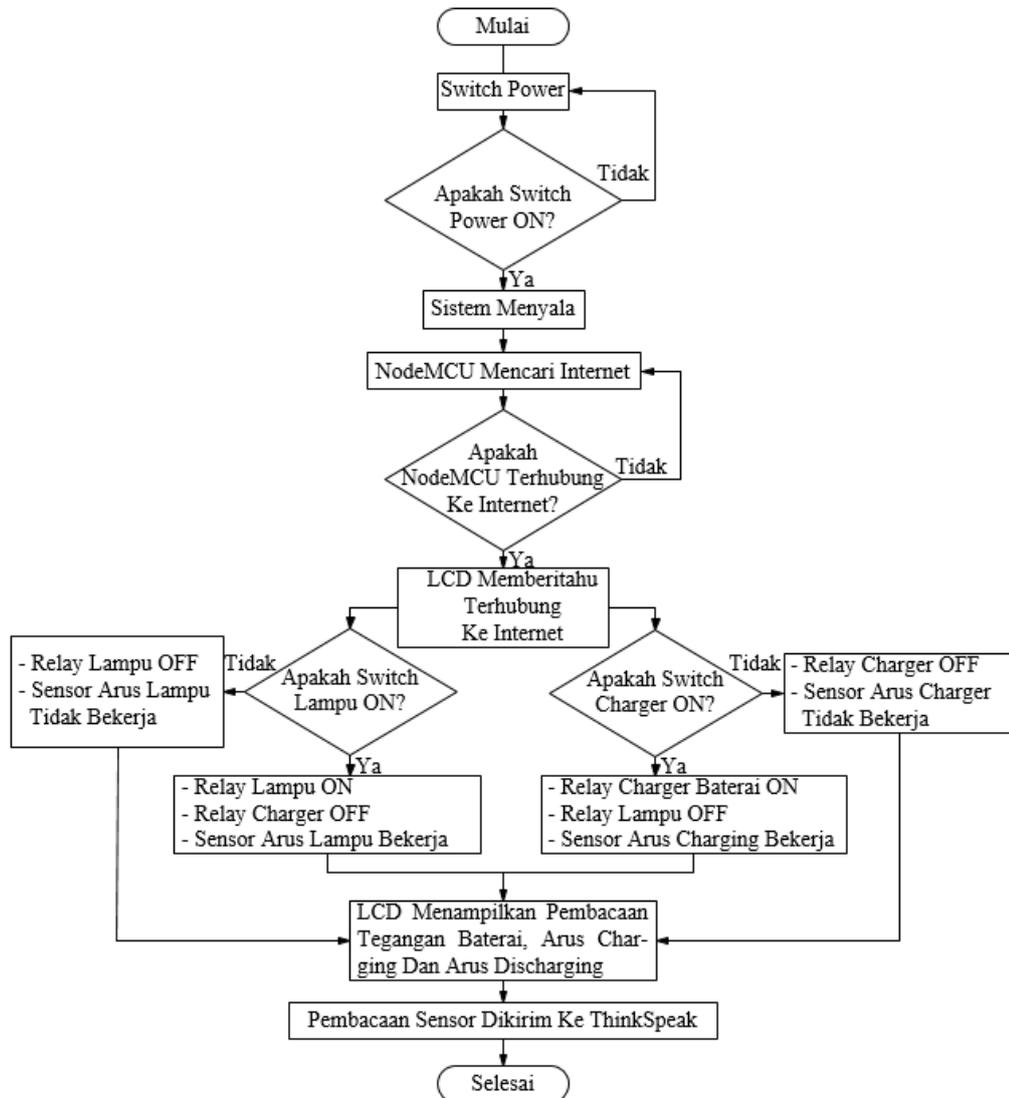
3.1.3.9 Pengujian Modul Relay

Pengujian modul relay bertujuan untuk mengetahui kesesuaian kerja relay dengan perintah yang diinput. Sebagai *otomatic switch*, relay harus memiliki keandalan dalam memutus dan menyambung sesuai perintah guna menghindari kerusakan alat akibat ketidaksesuaian aktuator (pada kondisi ini, aktuator yang dimaksud adalah modul relay) dengan perintah.

3.1.4 Perakitan Sistem

Perakitan sistem merupakan penggabungan dari seluruh komponen-komponen yang telah dikumpulkan guna membangun sebuah alat dengan sistem kerja sesuai harapan. Sistem yang dibangun merupakan sistem dengan arsitektur sistem pada Gambar 3.2 dan sesuai dengan kerja sistem pada Gambar 3.19.

3.1.5 Pengujian Sistem

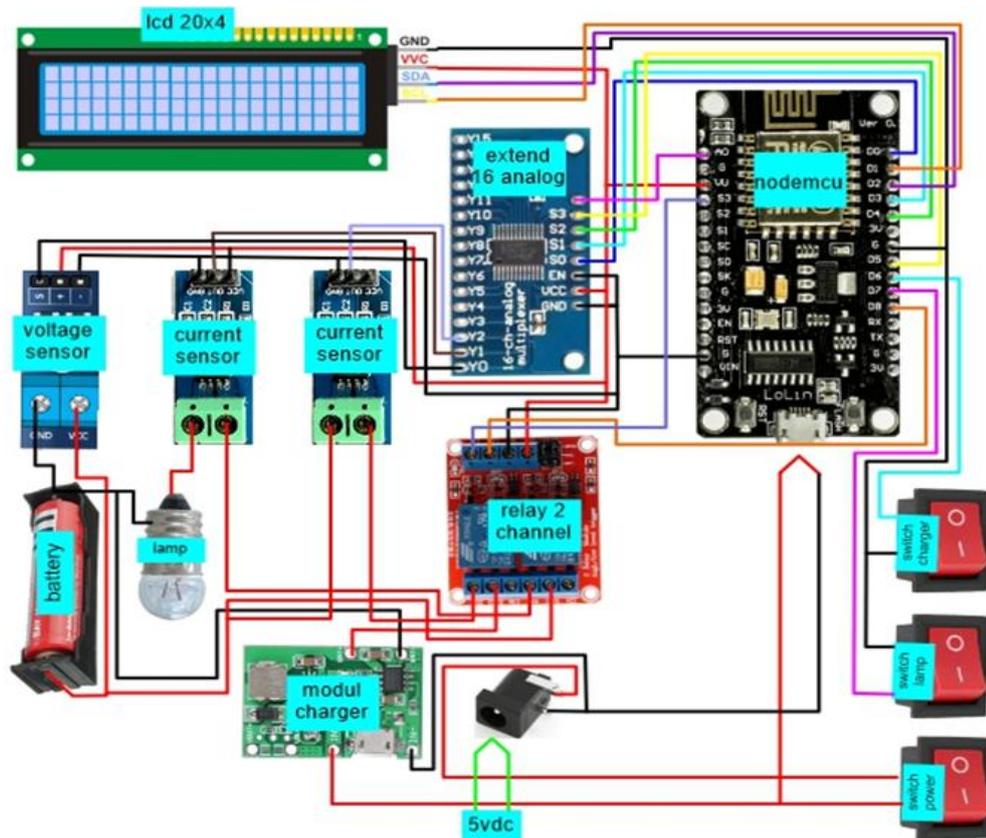


Gambar 3.19. *Flowchart* Pengujian Sistem

Gambar 3.19 menunjukkan *flowchart* pengujian sistem dari penelitian yang dilakukan ini. Pengujian sistem dilakukan guna mengetahui kesesuaian kerja alat dengan tujuan dilakukan penelitian ini. Dari *flowchart* yang ditunjukkan Gambar 3.19, alur pengujian sistem adalah sebagai berikut ini:

- Ketika *switch power ON*, maka sistem akan menyala.

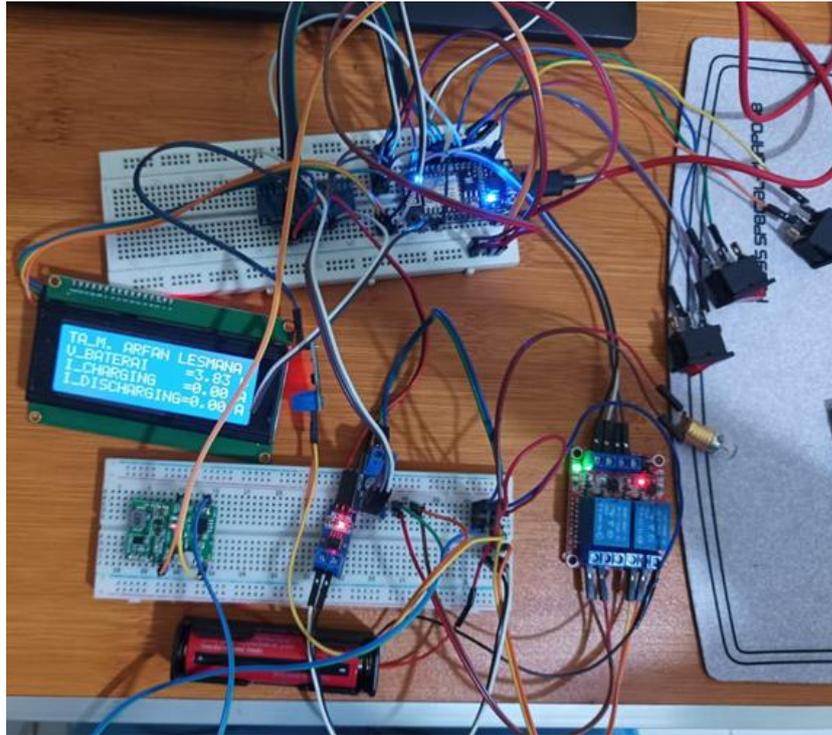
- Kemudian NodeMCU akan mencari jaringan internet agar terhubung ke sistem *internet of things*.
- Saat NodeMCU sudah terhubung dengan internet, LCD 20×4 akan menampilkan informasi tersebut.
- Ketika *switch* lampu ON, maka relay lampu akan ON / terhubung, sensor arus lampu akan mengukur arus *discharging* / arus lampu, relay *charger* baterai akan OFF.
- Ketika *switch charger* ON, relay *charger* akan ON, sensor *charger* akan membaca arus *charging*, dan relay lampu akan OFF.
- Saat proses *charging* maupun proses *discharging* berlangsung, sensor tegangan tetap berjalan mendeteksi besaran tegangan,
- Data dari masing-masing sensor akan dikirim ke *ThinkSpeak*.



Gambar 3.20. *Wiring* Pengujian Sistem

Gambar 3.20 menunjukkan *wiring* pengujian sistem yang merupakan *wiring* keseluruhan dimana telah tersambung dengan komponen-komponen untuk sistem *charging – discharging* baterai, sistem monitoring, dan sistem IoT. Relay 2 channel berfungsi sebagai penghindar proses *charging* dan *discharging* berlangsung dalam waktu yang bersamaan. Tiga buah saklar dengan masing-masing fungsi sebagai *switch power* (untuk menyalakan / mematikan seluruh sistem), *switch lamp* (untuk menyambung / memutuskan proses *discharging*), dan *switch charger* (untuk menyambung / memutuskan proses *charging*). NodeMCU sebagai penerima dan pengolah data juga sebagai penghubung sistem dengan IoT. *Multiplexer* sebagai penambah pin analog dan digital untuk NodeMCU. Modul *charger* untuk pengubah listrik AC menjadi listrik DC. LCD

20×4 sebagai penampil area lokal. Satu buah sensor tegangan DC untuk mengukur tegangan baterai. Dua buah sensor arus dengan masing-masing fungsi sebagai pengukur arus saat proses *charging* maupun saat proses *discharging* berlangsung.



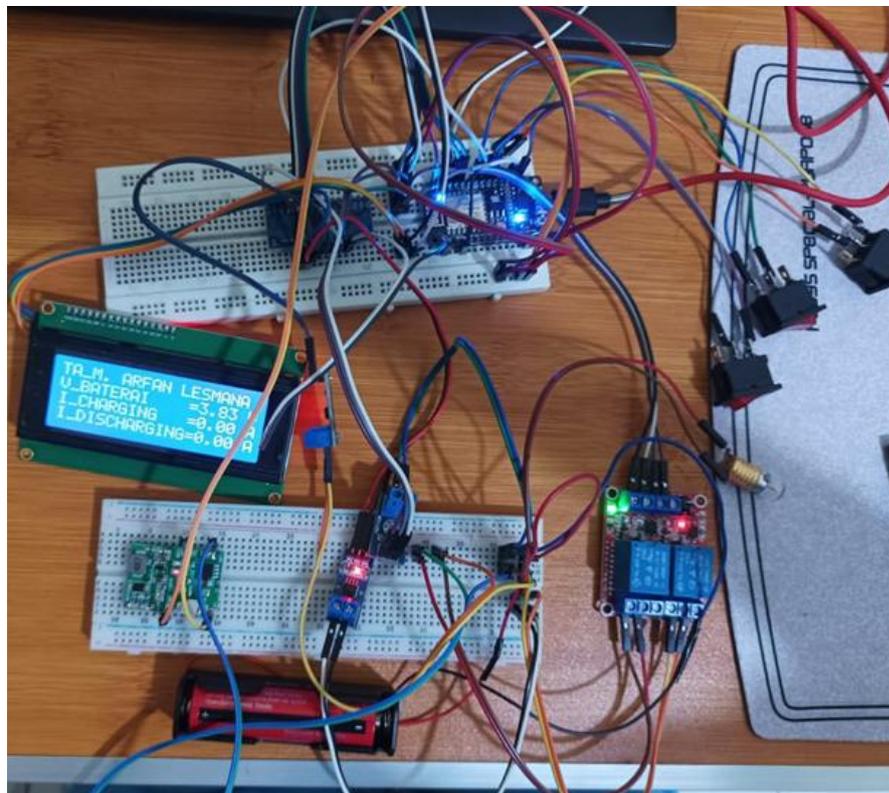
Gambar 3.21. Pengujian Sistem

Gambar 3.21 menunjukkan dokumentasi pengujian sistem dimana komponen-komponen disambungkan sesuai dengan yang ditunjukkan pada Gambar 3.20. Pengujian dilakukan guna mengetahui kesesuaian kerja pada sistem yang telah dibangun dengan *flowchart* pengujian sistem seperti yang ditunjukkan oleh Gambar 3.19.

3.1.6 Pengujian Sistem *Charging-Discharging* Baterai Li-Ion 3,7V

Pengujian ini dilakukan untuk mengetahui nilai tegangan dan arus saat proses *charging* maupun saat proses *discharging* berlangsung. Pengujian proses *charging* dilakukan dengan menghubungkan sumber listrik PLN dengan adaptor yang kemudian disambungkan dengan modul *charger* yang kemudian listrik keluaran modul *charger* ini akan dialirkan menuju baterai. Sehingga pada proses *charging* ini baterai berperan sebagai beban. Kemudian, proses *discharging* dilakukan dengan menghubungkan baterai dengan lampu dimana lampu yang digunakan berperan sebagai beban pada proses *discharging*. Pada kedua proses tersebut terdapat masing-masing sensor arus yang berbeda satu sama lain.

3.1.7 Pengujian Sistem Monitoring



Gambar 3.22. Pengujian Sistem Monitoring Pada LCD

Gambar 3.22 menunjukkan pengujian sistem monitoring dimana LCD menjadi penampil hasil monitoring yang dilakukan. Pengujian ini bertujuan untuk mengetahui kecakapan LCD dalam menampilkan data dari sensor tegangan dan sensor arus. Selain itu, pengujian monitoring bertujuan untuk mengetahui berhasil atau tidaknya sistem monitoring dalam menyajikan perubahan data secara *real time*.

3.1.8 Pengujian Sistem *Internet Of Things*

Pengujian sistem IoT bertujuan untuk mengetahui kemampuan NodeMCU untuk terkoneksi dengan internet dan terhubung ke *platform ThinkSpeak*. Pada pengujian ini dilakukan pengujian terhadap kesesuaian data yang dikirim oleh NodeMCU dengan data yang tampil pada tampilan *platform ThinkSpeak*.

3.1.9 Analisa Hasil Pengujian Sistem

Analisa hasil pengujian sistem dilakukan guna mengetahui dan mengkaji apakah hasil pengkajian sesuai atau tidak dengan sistem yang diharapkan. Jika hasil pengujian sesuai, maka dilanjutkan dengan kesimpulan. Namun, jika analisa hasil pengujian sistem tidak sesuai dengan apa yang telah direncanakan, maka dilakukan perakitan dan pengujian sistem ulang sampai mendapatkan hasil pengujian sistem yang sesuai.

3.2 Subjek Dan Objek Penelitian

Subjek yang digunakan adalah berupa sensor tegangan dan sensor arus sebagai *point result* dari sistem monitoring. Sedangkan objek penelitian yang

digunakan adalah baterai Li-Ion dengan tegangan 3,7V sebagai media penyimpan listrik.

3.3 Implementasi Rancangan Alat

Implementasi rancangan alat merupakan realisasi dari perancangan yang telah dikaji secara teoritis dan didesain secara fungsional. Hal tersebut bertujuan agar dapat mengetahui kesinambungan antara hasil kerja alat yang diteliti dengan teori yang berlaku.

3.4 Analisis Data

Analisis data dilakukan untuk mengkaji hasil penelitian dan menyesuaikan hasil tersebut dengan teori yang telah dikaji. Sehingga didapatkan hasil implementasi yang sesuai rancangan dan dapat dijelaskan secara ilmiah. Selain itu dari analisis data ini dapat diketahui kelebihan dan kekurangan kerja alat yang diteliti. Data yang akan dianalisa merupakan hasil implementasi beserta pengujiannya yang terdiri dari pengujian sistem *charging*, sistem *discharging*, sistem monitoring, dan sistem IoT.