

**PROTOTYPE DELIVERY ROBOT  
MENGUNAKAN NODEMCU ESP8266**

**LAPORAN KULIAH KERJA PRAKTEK  
PROGRAM STUDI D3 TEKNIK KOMPUTER**



**Oleh :**

**NPM  
43A87005190050**

**NAMA  
BAGUS ILHAM SANTOSO**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN  
KOMPUTER  
BANI SALEH  
BEKASI  
TAHUN 2022**

## **KATA PENGANTAR**

Puji dan syukur atas semua Rahmat dan Karunia-Nya yang Allah berikan kepada saya sehingga saya bisa menyelesaikan Laporan Kuliah Kerja Praktek ini yang berjudul “Prototype Delivery Robot Menggunakan NodeMCU ESP8266”.

Laporan Kuliah Kerja Praktek ini disusun untuk memenuhi persyaratan agar dapat menyelesaikan mata kuliah Kuliah Kerja Praktek pada jurusan Teknik Komputer.

Dalam Kesempatan ini dengan segala kerendahan hati, penulis sampaikan ucapan terima kasih yang tak terhingga kepada :

1. Bapak Domo Pranowo Kuswandono, ST., MT. Sekolah Tinggi Manajemen Informatika dan Komputer.
2. Bapak Rahmadi S.Kom., M.Kom., Sekolah Tinggi Manajemen Informatika dan Komputer.
3. Kepada kedua Orang Tuaku, Saudara – Saudaraku serta teman-teman yang telah memberikan kasih sayang, semangat, dukungan, serta doa kepada saya sehingga penulisan laporan ini dapat diselesaikan.

Penulis menyadari bahwa dalam penyusunan Laporan Kuliah Kerja Praktek ini masih banyak kekurangan yang baik disengaja maupun tidak disengaja, untuk itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun dari pembaca demi perbaikan dan kesempurnaan Laporan Kuliah Kerja Praktek. Akhir kata semoga penulisan dapat bermanfaat bagi semua pihak, khususnya bagi penulis dan bagi para pembaca umumnya.

Bekasi, Februari 2022

Penulis

Bagus Ilham Santoso

**LEMBAR PERSETUJUAN**  
**LAPORAN KULIAH KERJA PRAKTEK**  
**PROGRAM STUDI D3 TEKNIK KOMPUTER**  
**STMIK BANI SALEH**

**PROTOTYPE DELIVERY ROBOT**  
**MENGGUNAKAN NODEMCU ESP8266**

NPM  
43A87005190050

NAMA  
BAGUS ILHAM SANTOSO

Bekasi, Januari 2022

Dosen Pembimbing,

Domo Pranowo Kuswandono, ST., MT.

Ketua Program Studi  
D3 Teknik Komputer

Rahmadi S.Kom. M.Kom  
NIDN : 0431017502

## DAFTAR ISI

Kata Pengantar .....	i
Lembar Pengesahan .....	ii
Daftar Isi.....	iii
Daftar Gambar.....	iv
Daftar Tabel .....	vii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penulisan .....	1
1.3 Batasan Masalah .....	2
1.5 Sistematika Penulisan .....	2
BAB II LANDASAN TEORI .....	4
2.1 Sejarah Mikrokontroller .....	4
2.2 NodeMCU ESP8266 .....	7
2.2.1 Sejarah NodeMCU .....	8
2.3 Motor Driver L298N .....	8
2.3.1 Brief Data .....	9
2.3.2 Schematic Diagram Motor Drive L298N .....	10
2.3.3 Board Dimension & Pins Function .....	10
2.4 TT Motor DC & GearBox .....	11
2.4.1 Technical Details.....	12
2.4.2 Schematic Diagram TT Motor DC .....	12
2.5 Battery Holder .....	13
2.6 Rechargeable Battery Lithium 18650 .....	14
2.7 Rainbow Jumper Wire .....	15
2.8 Software Arduino IDE .....	16
2.8.1 Struktur Dasar Penulisan Sketch .....	16
2.8.2 Sybtaks dalam Penulisan Program .....	17
2.8.3 Fitur – Fitur di Software Arduino .....	17
2.9 Aplikasi BLYNK .....	18

BAB III Perancangan Metode Dan Schematics Alat Delivery Robot	
Menggunakan Nodemcu Esp8266 .....	20
3.1 Schematics Delivery Robot .....	20
3.2 Flowchart Alat .....	21
3.3 Blok Diagram Alat .....	22
3.4 Blok Diagram BLYNK Kontroller .....	22
BAB IV Perancangan .....	23
4.1 Persiapan Alat .....	24
4.2 Solder Kabel Pada TT Motor DC .....	24
4.3 Pemasangan TT Motor DC Ke Chasis Mobil .....	24
4.4 Pemasangan Kabel Motor DC pada Driver Motor .....	25
4.5 Pemasangan Kabel Driver Motor Ke ESP8266 .....	26
4.6 Pemrograman Alat .....	27
4.7 Tabel Pengujian .....	31
4.8 Gambar Alat .....	32
4.8.1 Tampak Depan .....	32
4.8.2 Tampak Belakang.....	33
4.8.3 Tampak Samping Kanan .....	33
4.8.4 Tampak Samping Kiri .....	34
4.8.5 Tampak Atas .....	34
BAB V Penutup .....	35
5.1 Kesimpulan .....	35
5.2 Saran.....	35
DAFTAR PUSTAKA	

## DAFTAR GAMBAR

Gambar 2.1 Mikrokontroler .....	5
Gambar 2.2 NodeMCU ESP8266 .....	7
Gambar 2.3 Motor Drive L298N .....	8
Gambar 2.3.2 Diagram Motor Driver L298N .....	10
Gambar 2.3.3 Board Dimension & Pins .....	10
Gambar 2.4 TT Motor DC & Gearbox .....	11
Gambar 2.4.4 Schematic Diagram TT Motor DC .....	12
Gambar 2.5 Battery Holder Dual Cell.....	13
Gambar 2.5.1 Battery Holder Specification .....	13
Gambar 2.6 Battery Lithium 18650 3.7V .....	14
Gambar 2.7 Kabel Jumper.....	15
Gambar 2.8 Arduino IDE .....	16
Gambar 2.9 BLYNK .....	18
Gambar 3.1 Schematics Delivery Robot.....	20
Gambar 3.1 Flowchar Alat.....	21
Gambar 3.3 Blok Diagram Alat .....	22
Gambar 3.4 Blok Diagram BLYNK Kontroller.....	22
Gambar 4.1 Persiapan Alat .....	23
Gambar 4.2 Solder Kabel pada TT Motor DC .....	24
Gambar 4.3 Pemasangan TT Motor DC ke Chasis Mobil .....	24
Gambar 4.4 Pin Motor OUT1 .....	25
Gambar 4.4 Pin Motor OUT2 .....	25
Gambar 4.4 Pin Motor OUT3 .....	25
Gambar 4.4 Pin Motor OUT4 .....	26
Gambar 4.5 Pemasangan Kabel Driver Motor ke ESP8266 .....	16
Gambar 4.8.1 Tampak Depan Mobil.....	32
Gambar 4.8.2 Tampak Belakang Mobil.....	33
Gambar 4.8.3 Tampak Samping Kanan .....	33

Gambar 4.8.4 Tampak Samping Kiri .....	34
Gambar 4.8.5 Tampak Atas .....	34

## **DAFTAR TABEL**

Tabel 4.7 Tebel Arus Listrk yang Diberikan Pada Mobil .....	31
Tabel 4.7 Tebel Arus Listrik Mobil Tidak Menyala .....	32
Tabel 4.7 Tebel Arus Listri Ketika Mobil Berjalan .....	32



## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Dalam masa industri saat ini banyak perusahaan ingin meningkatkan efisiensi pekerjaan yang dilakukan sumber daya manusia dengan cepat dan tepat. Semakin cepat pekerjaan yang dilakukan karyawan maka semakin banyak produktifitas pekerjaan yang yang disanggupi oleh karyawan.

Saat pekerjaan divisi sudah menyelesaikan produksi suatu barang, maka barang yang sudah selesai dikerjakan maka akan dikirimkan ke divisi lain untuk diproses lebih lanjut lagi. Ketika pengiriman barang tersebut maka dibutuhkan satu orang untuk mengirimkan barang yang sudah jadi ke divisi lainnya. Maka, dalam divisi produksi harus menggantikan seseorang yang mengirimkan barang ke divisi lain atau mengurangi produktifitas sementara untuk menunggu orang tersebut selesai mengirimkan barang yang akan dikirimkan ke divisi lainnya.

Kecepatan dan Ketepatan dalam produksi sangat diperhitungkan lebih, untuk memaksimalkan hasil dan jumlah yang banyak dalam proses produksi. Jika hasil produksi yang di hasilkan sangat banyak maka perusahaan menguntungkan penjualan yang lebih. Maka dari itu, Prototype Delivery Car Robot ini dibuat untuk menggantikan tenaga manusia dalam pengiriman antar divisi, sehingga dapat menghemat waktu dan memaksimalkan pekerjaan di dalam divisi tersebut.

#### **1.2 Tujuan Penulisan**

Tujuan pembuatan laporan Kuliah Kerja Praktek ini adalah untuk membuat rancangan sebuah alat transportasi yang dapat digunakan sebagai alat pengantar barang antar divisi yang menggunakan kontrol jarak jauh menggunakan aplikasi Android.

Manfaat penelitian ini adalah :

- a. Sebagai alat bantu untuk mengirimkan komponen/barang antar divisi dalam perusahaan.

- b. Memberikan kemudahan dan kecepatan untuk karyawan industri mengirimkan komponen/bahan produksi antar divisi.
- c. Untuk memenuhi mata kuliah KKP (Kuliah Kerja Praktek) jurusan D3 Teknik Komputer STMIK Bani Saleh.
- d. Mempraktikan dan menerapkan ilmu pengetahuan IOT yang didapat selama mengikuti perkuliahan.
- e. Mengembangkan kreatifitas bagi penulis dibidang Teknik Komputer sebagai bidang yang dipelajari.

### **1.3 Batasan Masalah**

- a. Mikrokontroler yang digunakan adalah ESP8266 NodeMCU Lolin.
- b. Bahasa Pemrograman yang digunakan pada Mikrokontroler yaitu Bahasa C.
- c. ESP8266 NodeMCU digunakan sebagai Access penghubung kontroler Blynk.
- d. Laporan penulisan ini berupa prototype delivery robot yang masih dikembangkan di lab.
- e. Prototype ini hanya berfungsi untuk mendukung pengiriman barang menggunakan aplikasi.

### **1.4 Sistematika Penulisan**

#### **BAB I : PENDAHULUAN**

Berisikan latar belakang permasalahan yang diangkat dari penulisan laporan ini

#### **BAB II : LANDASAN TEORI**

Penjelasan tentang pengertian Mikrokontroller dan Sejarah Mikrokontroller serta tentang teori pengertian spesifikasi alat yang digunakan pada laporan ini.

**BAB III : PERANCANGAN METODE ALAT**

Penjelasan rancangan metode alat dan schematic yang digunakan pada laporan penulisan ini.

**BAB IV : PERANCANGAN ALAT**

Perancangan alat dan pembahasan dari source code yang digunakan pada penulisan laporan ini.

**BAB V : KESIMPULAN DAN SARAN**

Kesimpulan yang diperoleh dari penulisan laporan KKP ini beserta beberapa saran untuk pengembangan alat pada laporan penulisan ini.

**DAFTAR PUSTAKA**

## **BAB II**

### **LANDASAN TEORI**

Mikrokontroler merupakan sebuah chip atau IC yang didalamnya sudah terdapat prosesor (ALU, CU & Register), memori dan peripheral tambahan lain. Semua bagian tersebut dipadatkan dalam satu buah chip. Banyaknya peripheral yang ada dalam mikrokontroler tergantung dari masing-masing tipe dan spesifikasi pabrik. Tidak bisa disamakan isi dari mikrokontroler A dengan mikrokontroler B. Karena mikrokontroler sendiri dirancang untuk spesifikasi kebutuhan yang bermacam-macam. (Erwin Techable n.d.)

Bahasa pemrograman yang digunakan adalah Bahasa C yang merupakan bahasa tingkat tinggi dan umum yang bertujuan membuat suatu proyek dan bisa dipakai dalam sehari-hari. Bahasa pemrograman C masih dipakai saat ini karena mudah dipelajari dengan sumbernya yang sangat banyak dan semakin berkembang di berbagai macam platform seperti Windows, Linux, MacOS dan Android. Bahasa C dalam dijalankan menggunakan teks editor IDE (Integrated Development Environment) agar dapat dimengerti oleh komputer.

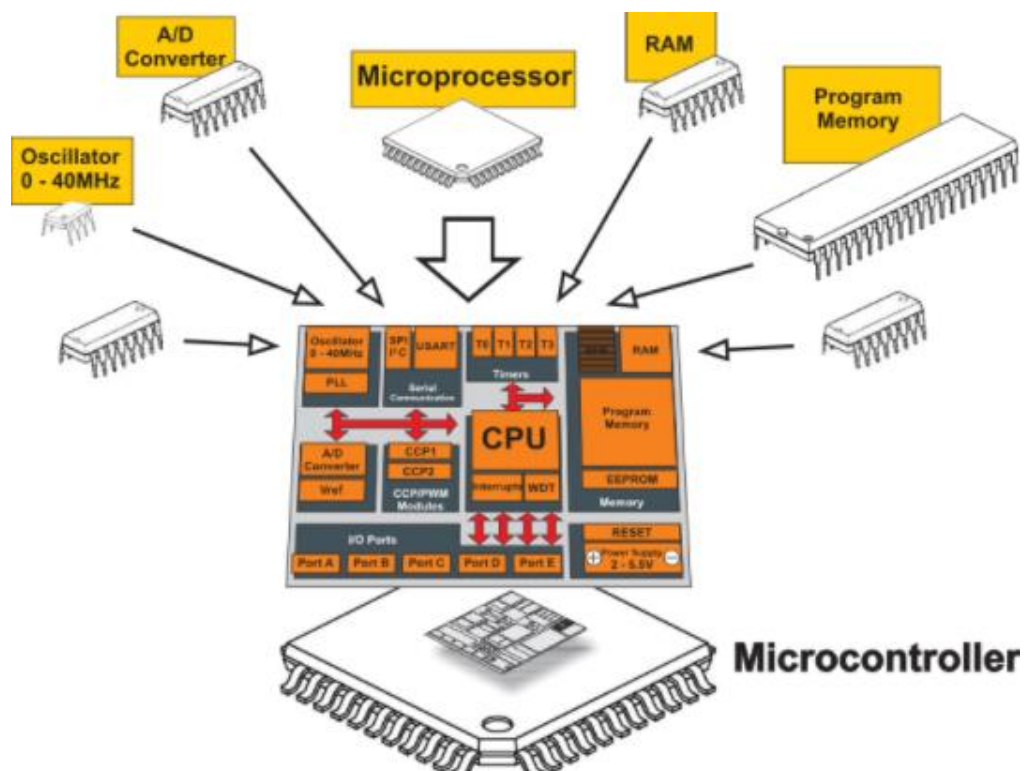
Perancangan robot Delivery Robot dengan menggunakan Mikrokontroller ESP8266 perlu pengetahuan dan teori tentang mikrokontroler yang digunakan, komunikasi data, aktuator serta media yang digunakan untuk dapat mengakses ke dalam hardware. Berikut ini adalah landasan teori berdasarkan Datasheet yang digunakan untuk menyelesaikan Kuliah Kerja Praktek tersebut :

#### **2.2.1 Sejarah Mikrokontroller**

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output.

Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca

dan menulis data. Sekedar contoh, bayangkan diri Anda saat mulai belajar membaca dan menulis, ketika Anda sudah bisa melakukan hal itu Anda bisa membaca tulisan apapun baik buku, cerpen, artikel dan sebagainya, dan Anda pun bisa pula menulis hal-hal sebaliknya. Begitu pula jika Anda sudah mahir membaca dan menulis data maka Anda dapat membuat program untuk membuat suatu sistem pengaturan otomatis menggunakan mikrokontroler sesuai keinginan Anda. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini. (IMMERSA LAB 2014)



Gambar 2.1 Mikrokontroler

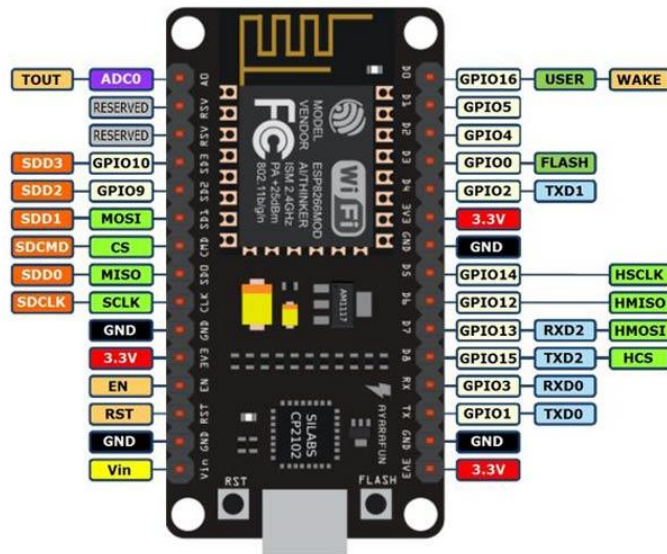
Sejarah mikrokontroler dan mikroprosesor dapat ditelusuri kembali ke penemuan MOSFET (transistor efek lapangan logam-oksid-semikonduktor), juga dikenal sebagai transistor MOS.

- a) Transistor MOS ditemukan oleh Mohamed M. Atalla dan Dawon Kahng di Bell Labs pada tahun 1959, dan pertama kali ditunjukkan pada tahun 1960.
- b) Pada tahun yang sama, Atalla mengusulkan konsep sirkuit terintegrasi MOS, yang merupakan chip sirkuit terintegrasi yang dibuat dari MOSFETs.
- c) Pada tahun 1964, chip MOS telah mencapai kepadatan transistor yang lebih tinggi dan biaya manufaktur yang lebih rendah daripada chip bipolar. Chip MOS semakin meningkat dalam kompleksitas pada tingkat yang diprediksi oleh hukum Moore, yang mengarah ke integrasi skala besar (LSI) dengan ratusan transistor pada satu chip MOS pada akhir 1960-an. Penerapan chip MOS LSI untuk komputasi adalah dasar untuk mikroprosesor pertama, karena insinyur mulai mengakui bahwa prosesor komputer lengkap dapat terkandung pada satu chip LSI MOS.

Mikroprosesor multi-chip pertama, Four-Phase Systems AL1 pada tahun 1969 dan Garrett AiResearch MP944 pada tahun 1970, dikembangkan dengan beberapa chip LSI MOS. Mikroprosesor chip tunggal pertama adalah Intel 4004, dirilis pada satu chip MOS LSI pada tahun 1971. Ini dikembangkan oleh Federico Faggin, menggunakan teknologi MOS silikon-gate-nya, bersama dengan insinyur Intel Marcian Hoff dan Stan Mazor, dan insinyur Busicom Masatoshi Shima.

- d) Diikuti oleh Intel 4040 4-bit, Intel 8008 8-bit, dan Intel 8080 8-bit. Semua prosesor ini memerlukan beberapa chip eksternal untuk menerapkan sistem kerja, termasuk chip antarmuka memori dan periferal. Akibatnya, total biaya sistem adalah beberapa ratus (1970-an dolar AS), sehingga tidak mungkin untuk komputerisasi peralatan kecil secara ekonomi. MOS Technology memperkenalkan mikroprosesor sub-\$ 100, 6501 dan 6502, dengan tujuan utama untuk mengatasi hambatan ekonomi ini, tetapi mikroprosesor ini masih membutuhkan dukungan eksternal, memori, dan chip periferal yang menjaga total biaya sistem dalam ratusan dolar. (Tekno Info Kamu 2021)

## 2.2 NodeMCU ESP8266



Gambar 2.2 NodeMCU ESP8266

NodeMCU adalah sebuah platform IoT yang bersifat opensource. Terdiri dari perangkat keras berupa System On Chip ESP8266 dari ESP8266 buatan Espressif System, juga firmware yang digunakan, yang menggunakan bahasa pemrograman scripting Lua. Istilah NodeMCU secara default sebenarnya mengacu pada firmware yang digunakan daripada perangkat keras development kit.

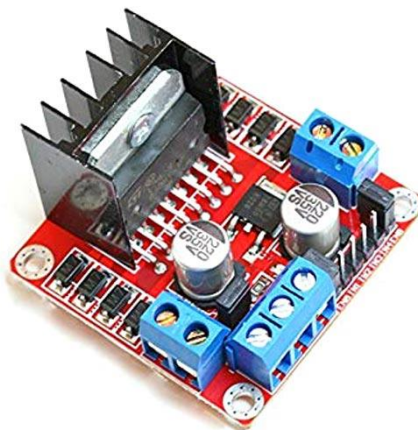
NodeMCU bisa dianalogikan sebagai board arduino-nya ESP8266. Dalam seri tutorial ESP8266 embeddednesia pernah membahas bagaimana memprogram ESP8266 sedikit merepotkan karena diperlukan beberapa teknik wiring serta tambahan modul USB to serial untuk mengunduh program. Namun NodeMCU telah me-package ESP8266 ke dalam sebuah board yang kompak dengan berbagai fitur layaknya mikrokontroler + kapabilitas akses terhadap Wifi juga chip komunikasi USB to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB persis yang digunakan sebagai kabel data dan kabel charging smartphone Android.

### 2.2.1 Sejarah NodeMCU

Sejarah lahirnya NodeMCU berdekatan dengan rilis ESP8266 pada 30 Desember 2013, Espressif Systems selaku pembuat ESP8266 memulai produksi ESP8266 yang merupakan SoC Wi-Fi yang terintegrasi dengan prosesor Tensilica Xtensa LX106. Sedangkan NodeMCU dimulai pada 13 Oktober 2014 saat Hong me-commit file pertama nodemcu-firmware ke Github. Dua bulan kemudian project tersebut dikembangkan ke platform perangkat keras ketika Huang R meng-commit file dari board ESP8266 , yang diberi nama devkit v.0.9.

Berikutnya, di bulan yang sama. Tuan PM memporting pustaka client MQTT dari Contiki ke platform SOC ESP8266 dan di-commit ke project NodeMCU yang membuatnya mendukung protokol IOT MQTT melalui Lua. Pemutakhiran penting berikutnya terjadi pada 30 Januari 2015 ketika Devsaurus memporting u8glib ke project NodeMCU yang memungkinkan NodeMCU bisa mendrive display LCD, OLED, hingga VGA. Demikianlah, project NodeMCU terus berkebang hingga kini berkat komunitas open source dibalikinya, pada musim panas 2016 NodeMCU sudah terdiri memiliki 40 modul fungsionalitas yang bisa digunakan sesuai kebutuhan developer. (Saputro 2007)

### 2.3. Motor Driver L298N



*Gambar 2.3 Motor Drive L298N*



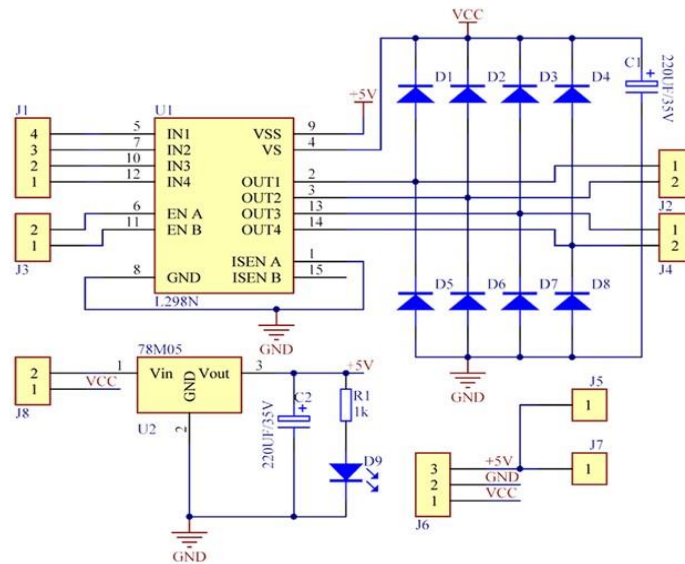
Driver Motor L298N adalah sebuah modul yang sering sekali digunakan untuk mengendalikan motor DC. Dengan menggunakan Driver Motor L298N kita bisa dengan mudah mengendalikan baik itu kecepatan maupun arah rotasi 2 motor sekaligus. Driver Motor L298N dirancang menggunakan IC L298 Dual H-Bridge Motor Driver berisikan gerbang gerbang logika yang sudah sangat populer dalam dunia elektronika sebagai pengendali motor. (Al Khairi 2021)

### 2.3.1. Brief Data

- a) Input Voltage: 3.2V~40Vdc
- b) Driver: L298N Dual H Bridge DC Motor Driver
- c) Power Supply: DC 5 V - 35 V
- d) Peak current: 2 Amp
- e) Operating current range: 0 ~ 36mA \Control signal input voltage range :
  - Low:  $-0.3V \leq V_{in} \leq 1.5V$ .
  - High:  $2.3V \leq V_{in} \leq V_{ss}$ .
- f) Enable signal input voltage range :
  - o Low:  $-0.3 \leq V_{in} \leq 1.5V$  (control signal is invalid).
  - o High:  $2.3V \leq V_{in} \leq V_{ss}$  (control signal active).
- g) Maximum power consumption: 20W (when the temperature  $T = 75^{\circ}C$ ).
- h) Storage temperature:  $-25^{\circ}C \sim +130^{\circ}C$ .
- i) On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- j) Size: 3.4cm x 4.3cm x 2.7cm

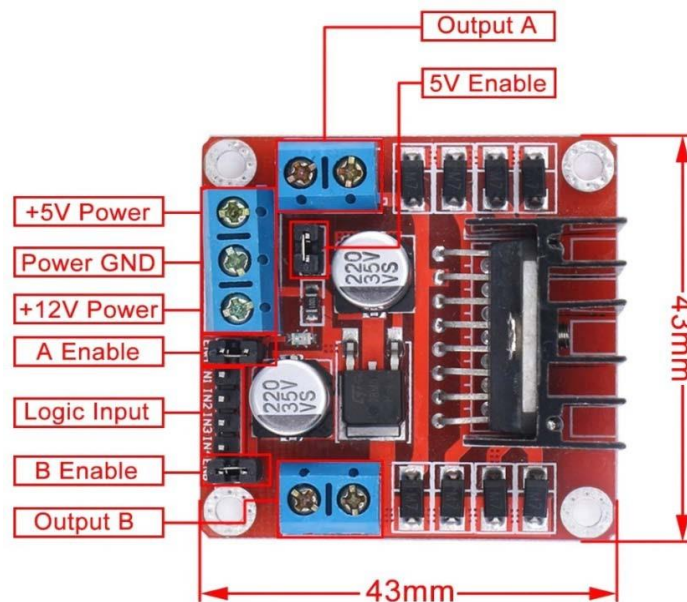
(HandsOn Tech 2002)

### 2.3.2 Schematic Diagram Motor Driver L298N



Gambar 2.3.2 Diagram Motor Driver L298N

### 2.3.3 Board Dimension & Pins Function



Gambar 2.3.3 Board Dimension & Pins

#### 2.4. TT Motor DC & GearBox



*Gambar 2.4 TT Motor DC & Gearbox*

Motor gearbox plastik tahan lama ini (juga dikenal sebagai motor 'TT') adalah cara yang mudah dan murah untuk membuat proyek Anda bergerak. Ini adalah Motor Gearbox DC TT dengan rasio roda gigi 1:48, dan dilengkapi dengan kabel 2 x 200mm dengan konektor laki-laki 0.1" yang cocok untuk papan tempat memotong roti. Sempurna untuk dicolokkan ke papan tempat memotong roti atau blok terminal.

Anda dapat memberi daya pada motor ini dengan 3VDC hingga 6VDC, tentu saja motor ini akan bekerja sedikit lebih cepat pada tegangan yang lebih tinggi. Kami mengambil satu motor dan menemukan statistik ini saat menjalankannya dari suplai bench-top :

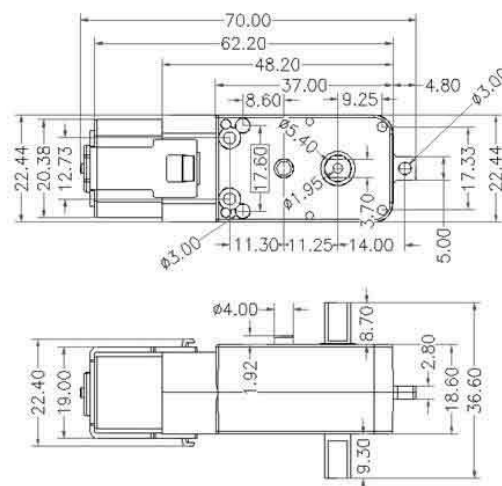
- a) Pada 3VDC kami mengukur 150mA @ 120 RPM tanpa beban, dan 1,1 Amps saat terhenti.
- b) Pada 4.5VDC kami mengukur 155mA @ 185 RPM tanpa beban, dan 1,2 Amps saat terhenti
- c) Pada 6VDC kami mengukur 160mA @ 250 RPM tanpa beban, dan 1,5 Amps saat terhenti.

Perhatikan bahwa ini adalah motor yang sangat dasar, dan tidak memiliki enkoder bawaan, kontrol kecepatan, atau umpan balik posisi. Tegangan masuk, rotasi padam! Akan ada variasi dari motor ke motor, jadi sistem umpan balik terpisah diperlukan jika Anda membutuhkan gerakan presisi.

#### 2.4.1 Technical Details

- Rated Voltage: 3~6V
- Continuous No-Load Current: 150mA +/- 10%
- Min. Operating Speed (3V): 90+/- 10% RPM
- Min. Operating Speed (6V): 200+/- 10% RPM
- Stall Torque (3V): 0.4kg.cm
- Stall Torque (6V): 0.8kg.cm
- Gear Ratio: 1:48
- Body Dimensions: 70 x 22 x 18mm
- Wires Length: 200mm & 28 AWG
- Weight: 30.6g

#### 2.4.2 Schematic Diagram TT Motor DC



Gambar 2.4.4 Schematic Diagram TT Motor DC

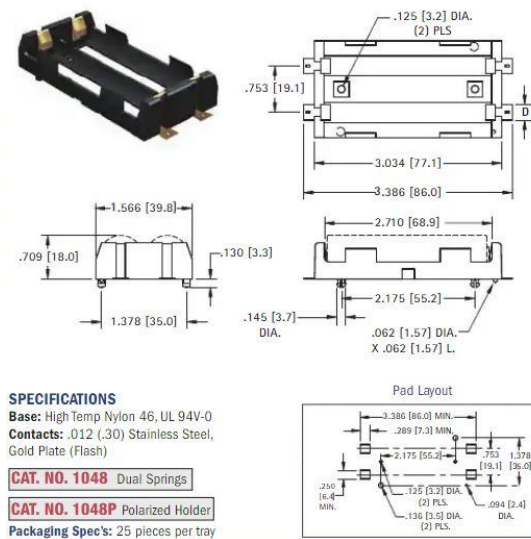
## 2.5. Battery Holder



Gambar 2.5 Battery Holder Dual Cell

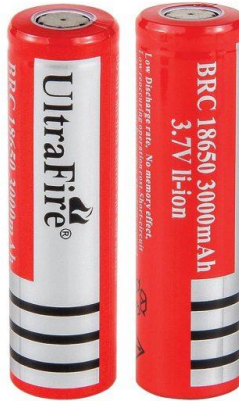
- **Base** : High Temp Nylon 46, UL 94V-0
- **Contacts** : 0.12 (.30) Stainless Steel, Gold Plate (Flash)

### 2.5.1 Battery Holder Specification



Gambar 2.5.1 Battery Holder Specification

## 2.6. Rechargeable Battery Lithium 18650



*Gambar 2.6 Battery Lithium 18650 3.7V*

Rechargeable Battery UltraFire 18650 Li-ion sangat populer di kalangan pelanggan. UltraFire memproduksi model yang berbeda dengan berbagai kapasitas baterai 18650.

Ada seri 18650 baterai li-ion isi ulang dari UltraFire. Kapasitas baterai bervariasi dari 2200mAh hingga 3400mAh. UltraFire juga memproduksi produk yang ditingkatkan dengan desain baru, jadi kami memiliki SKU yang berbeda untuk baterai 18650 yang memiliki kapasitas yang sama, seperti yang 3000mAh. Namun, kami tidak memproduksi baterai 18650 yang berkapasitas 4000mAh, 5000mAh atau lebih. Ukuran chip baterai li-ion 18650 terbatas sehingga tidak ada cara untuk mendapatkan kapasitas sebesar itu.

Banyak pelanggan yang menanyakan apakah baterai 18650 yang mereka beli asli atau tidak. Berikut adalah beberapa saran sederhana untuk Anda: Jika memungkinkan, selalu beli baterai dari situs web resmi UltraFire atau dari toko resmi Amazon kami. Baterai UltraFire 18650 menggunakan chip li-ion Sanyo, karena ukuran baterai 18650, kapasitas tertinggi kurang dari 4000mAh. Jadi jika melihat baterai 18650 yang memiliki kapasitas lebih besar dari 4000mAh, bisa dipastikan itu adalah baterai palsu, bukan dari UltraFire.

Baterai UltraFire 18650 menggunakan chip li-ion Sanyo, yang memberikan baterai kami kualitas tinggi dan masa pakai yang lama. Baterai 3000mAh 18650 dapat bekerja hingga 5-6 jam setelah terisi penuh. Ini memberi Anda cukup waktu untuk mendedikasikan diri dalam pekerjaan Anda. Dengan pengisi daya UltraFire yang sesuai, baterai 18650 dapat diisi dengan cepat dalam waktu singkat. Misalnya, dengan menggunakan pengisi daya 1000mAh, baterai li-ion dapat menyelesaikan pengisian daya dalam 1 setengah jam. (Ultra Fire 2018)

## 2.7. Rainbow Jumper Wire



Gambar 2.7 Kabel Jumper

### 2.7.1. Technical Details

- a) Length : 30 CM
- b) Weight : 1 Gram
- c) Operating temperature : -40 - 85°C
- d) Operating humidity : 20 % to 80%
- e) Storage temperature : -40 - 85°C
- f) Storage humidity : 15% to 90%

## 2.8. Software Arduino IDE



*Gambar 2.8 Arduino IDE*

Perangkat Lunak Arduino (IDE) open-source memudahkan penulisan kode dan mengunggahnya ke papan. Perangkat lunak ini dapat digunakan dengan papan Arduino apa pun. (arduino.cc n.d.)

Arduino IDE adalah software yang digunakan untuk membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada board yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke board yang ditentukan, dan meng-coding program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan library C/C++(wiring), yang membuat operasi input/output lebih mudah.

### 2.8.1 Struktur Dasar Penulisan Sketch

Setiap program arduino (biasa disebut sketch) mempunyai dua buah fungsi yang harus ada dalam setiap program yaitu :

#### a) **Void Setup (){}**

Void setup merupakan fungsi yang hanya menjalankan program yang ada didalam kurung kurawal sebanyak 1 kali.

#### b) **Void Loop () {}**

Fungsi ini akan dijalankan setelah setup (fungsi void setup) selesai, setelah dijalankan 1 kali, fungsi ini akan dijalankan lagi dan lagi secara terus menerus sampai catu daya (power) dilepaskan.



## 2.8.2 Sintaks Dalam Penulisan Program

### a) // (komentar 1 baris)

Digunakan untuk memberi komentar atau catatan pada kode-kode yang dibuat.

### b) /\* \*/ (komentar 2 baris)

Untuk menuliskan catatan pada beberapa baris sebagai komentar.

### c) {} (kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir serta digunakan juga pada fungsi dan pengulangan.

### d) ; (titik koma)

Setiap baris kode harus diakhiri dengan tanda ; (titik koma), jika ada titik koma yang hilang maka program tidak akan berjalan. (erintafifah 2021)

## 2.8.3 Fitur – Fitur di Software Arduino IDE

- 1) **Verify**, pada versi sebelumnya dikenal dengan istilah Compile. Sebelum aplikasi di-upload ke board Arduino, biasanya untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses Verify / Compile mengubah sketch ke binary code untuk di-upload ke mikrokontroler.
- 2) **Upload**, tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun kita tidak mengklik tombol verify, maka sketch akan di-compile, kemudian langsung diupload ke board. Berbeda dengan tombol verify yang hanya berfungsi untuk memverifikasi source code saja.
- 3) **New Sketch**, Membuka window dan membuat sketch baru.
- 4) **Open Sketch**, Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino

- 5) **Save Sketch**, menyimpan sketch, tapi tidak disertai dengan mengcompile.
- 6) **Serial Monitor**, Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.
- 7) **Keterangan Aplikasi**, pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal Compiling dan Done Uploading ketika kita mengcompile dan mengupload sketch ke board Arduino
- 8) **Konsol log**, Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika ada kesalahan pada sketch yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.
- 9) **Baris Sketch**, bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.
- 10) **Informasi Board dan Port**, Bagian ini menginformasikan port yang dipakai oleh board Arduino. (allgoblog.com 2017)

## 2.9 Aplikasi BLYNK



Gambar 2.9 BLYNK

BLYNK adalah platform untuk aplikasi OS Mobile (iOS dan Android) yang bertujuan untuk kendali module Arduino, Raspberry Pi, ESP8266, WEMOS D1, dan module sejenisnya melalui Internet.

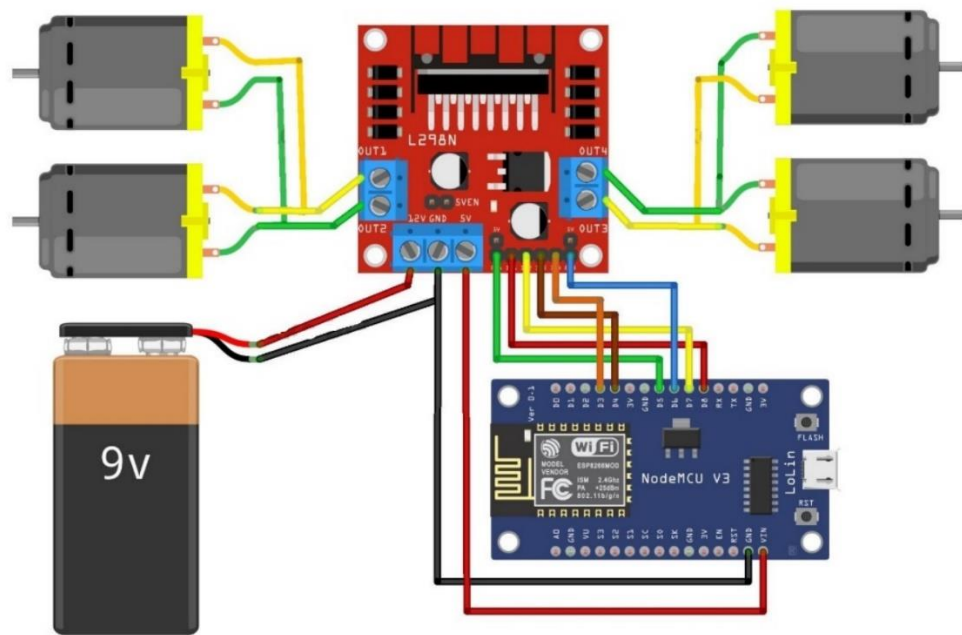
Aplikasi ini merupakan wadah kreatifitas untuk membuat antarmuka grafis untuk proyek yang akan diimplementasikan hanya dengan metode drag and drop widget.

Penggunaannya sangat mudah untuk mengatur semuanya dan dapat dikerjakan dalam waktu kurang dari 5 menit. Blynk tidak terikat pada papan atau module tertentu. Dari platform aplikasi inilah dapat mengontrol apapun dari jarak jauh, dimanapun kita berada dan waktu kapanpun. Dengan catatan terhubung dengan internet dengan koneksi yang stabil dan inilah yang dinamakan dengan sistem Internet of Things (IOT). (Nyebarilmu.com 2017)

## BAB III

### PERANCANGAN METODE DAN SCHEMATICS ALAT DELIVERY ROBOT MENGGUNAKAN NODEMCU ESP8266

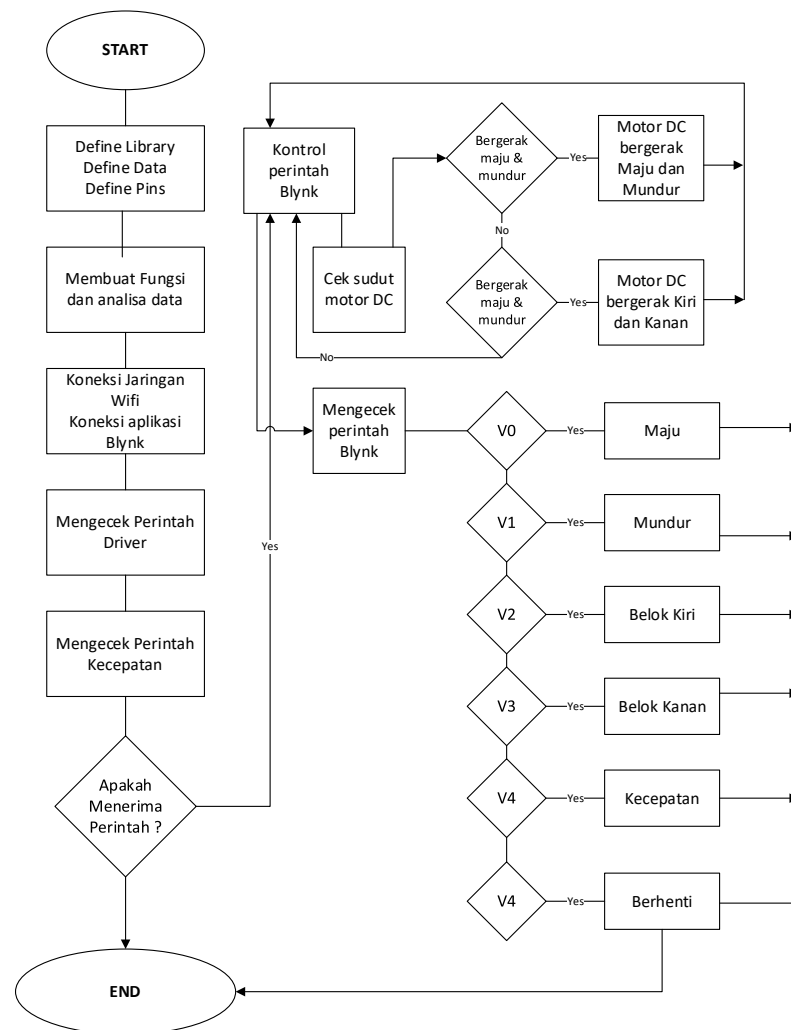
#### 3.1. Schematics Delivery Robot



*Gambar 3.1 Schematics Delivery Robot*

Ini adalah gambar schematic untuk alat Delivery Robot dengan menggunakan NodeMCU ESP8266 yang didesain menggunakan aplikasi Fritzing. Dalam perancangan ini menggunakan dua DC Motor melalui alat L298N Motor Driver. Melalui riset yang saya analisa motor driver L298N dapat berjalan dalam tegangan 5V sampai dengan 35V tegangan DC dengan besar maksimal daya yang ditampung yaitu 25W.

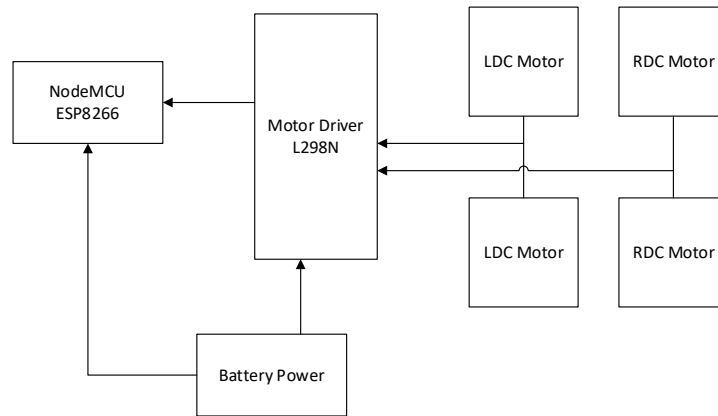
### 3.2. Flowchart Alat



Gambar 3.1 Flowchar Alat

User akan meng-upload perintah program kepada mikrokontroller untuk dapat mendefinisikan library sesuai dengan kebutuhan perangkat yang digunakan lalu, membuat fungsi untuk alat berjalan dengan semestinya dan mengkonfigurasi mikrokontroller dapat mendapatkan akses jaringan lokal. Jika sudah terupload maka ketika alat dijalankan akan mengecek perintah driver dan kecepatan default. Perintah eksekusi output dapat diberikan jika ada perintah maju, mundur, belok kiri dan belok kanan. Jika tidak ada, maka alat akan berhenti bergerak dan akan menunggu perintah.

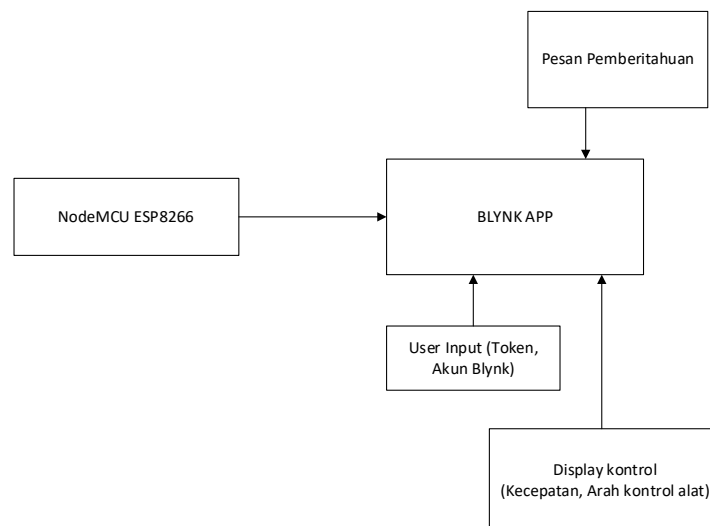
### 3.3. Blok Diagram Alat



Gambar 3.3 Blok Diagram Alat

Mikrokontroler ini akan mengolah data yang sudah diupload user sehingga dapat berkomunikasi dengan perangkat lainnya untuk memberikan perintah-perintah output. Power battery akan dibagi 2 jalur untuk Motor driver L298N dan NodeMCU ESP8266.

### 3.4. Blok Diagram BLYNK Kontroller



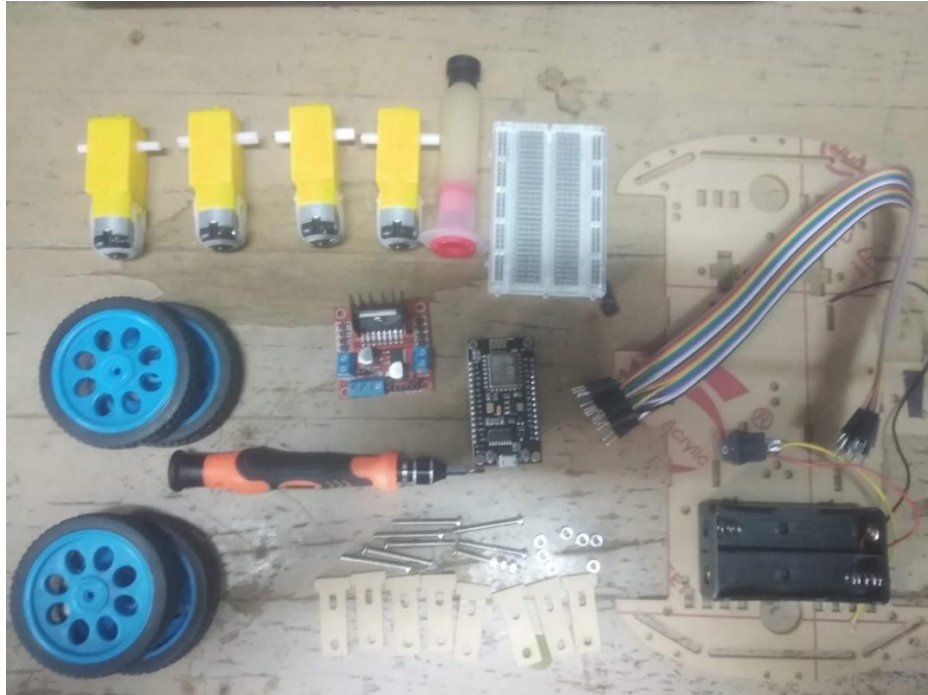
Gambar 3.4 Blok Diagram BLYNK Kontroller

Aplikasi BLYNK akan memproses authorisasi dari token yang diinput oleh user untuk mengkoneksikan dengan alat NodeMCU ESP8266 dan user akan membuat display kontrol beserta mengkonfigurasi display kontrol yang akan digunakan user untuk menggerakan alat.

## BAB IV

### PERANCANGAN

#### 4.1 Persiapan Alat

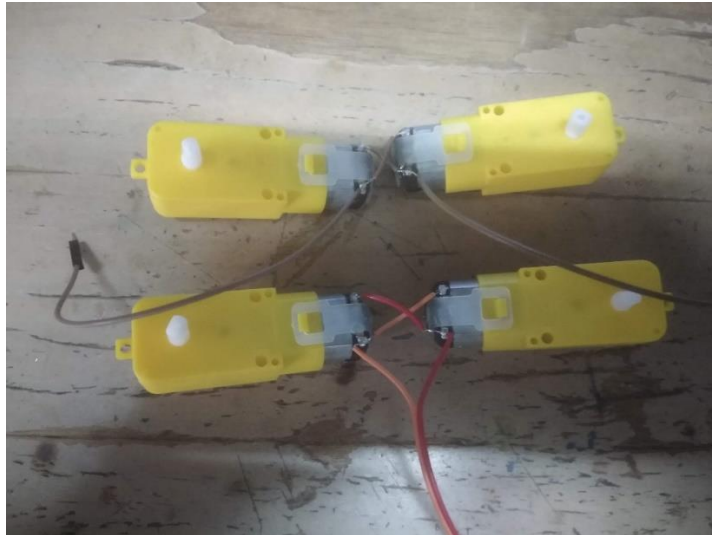


*Gambar 4.1 Persiapan Alat*

Persediaan Komponen :

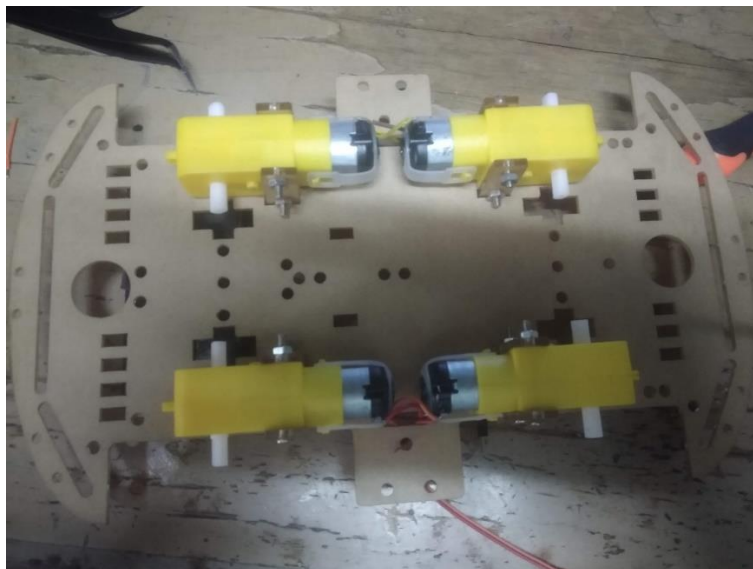
- 1 Buah NodeMCU ESP8266
- 1 Buah L298N Motor Driver
- 4 Buah TT Motor DC
- 4 Buah Rubber Tire / Ban Karet
- 1 Set Chasis Mobil
- 1 Buah Breadboard
- Kabel Jumper (Male to Male, Female to Male, Female to Female)

#### 4.2 Solder Kabel Pada TT Motor DC



*Gambar 4.2 Solder Kabel pada TT Motor DC*

#### 4.3 Pemasangan TT Motor DC Ke Chasis Mobil

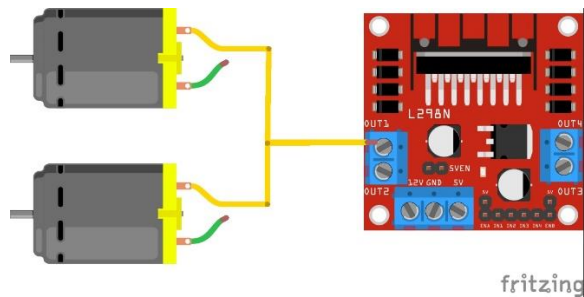


*Gambar 4.3 Pemasangan TT Motor DC ke Chasis Mobil*



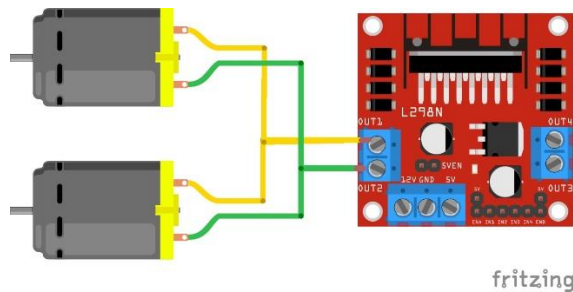
#### 4.4 Pemasangan Kabel Motor DC pada Driver Motor

##### 1) Pin Motor OUT1



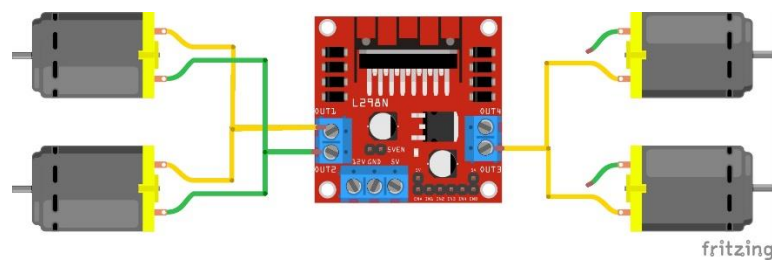
Gambar 4.4 Pin Motor OUT1

##### 2) Pin Motor OUT2



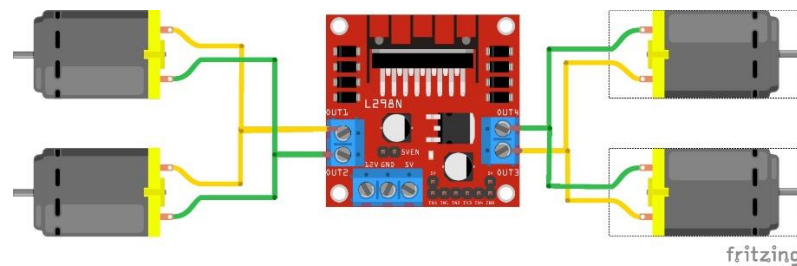
Gambar 4.4 Pin Motor OUT2

##### 3) Pin Motor OUT3



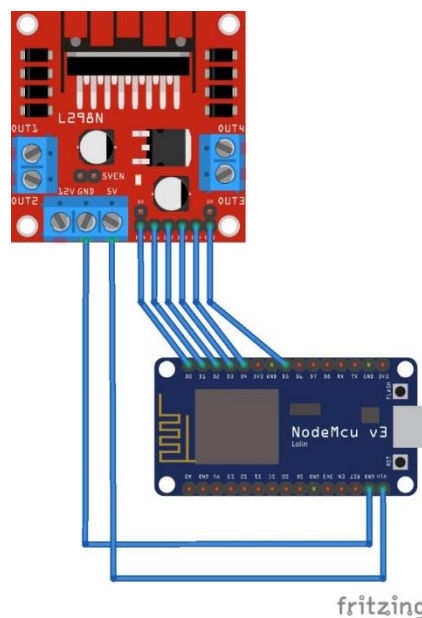
Gambar 4.4 Pin Motor OUT3

#### 4) Pin Motor OUT4



Gambar 4.4 Pin Motor OUT4

### 4.5 Pemasangan Kabel Driver Motor Ke ESP8266



Gambar 4.5 Pemasangan Kabel Driver Motor ke ESP8266

Pin ENA	→	D0
Pin ENB	→	D5
Pin IN1	→	D1
Pin IN2	→	D2
Pin IN3	→	D3
Pin IN4	→	D4

#### 4.6 Pemrograman Alat

*#define BLYNK\_PRINT Serial* -> Mendefinisikan controller Blynk

*#include <ESP8266WiFi.h>* → Menggunakan Library ESP8266

*#include <BlynkSimpleEsp8266.h>* → Menggunakan Library Blynk untuk ESP8266

*#define ENA D0*

*#define IN1 D1*

*#define IN2 D2*

*#define IN3 D3*

*#define IN4 D4*

*#define ENB D5*

Mendefinikas Pin Motor yang ada pada Driver Motor L298N

*bool forward = 0;*

*bool backward = 0;*

*bool left = 0;*

*bool right = 0;*

*int Speed;*

Mendeklarasikan bahwa semua perintah akan dimulai dari nol untuk mentiadakan sebuah value yang bersifat lain.

*char auth[] = "Mq9JcRJhCmaDvYuoJzVJ5BD2J6KI5vtZ";* //Enter your Blynk application auth token

Memasukkan Token yang diberikan dari aplikasi Blynk untuk mendapatkan koneksi antar Blynk yang sebagai controller alat dan Robot Mobil

*char ssid[] = "Bataz";* //Enter your WIFI name

*char pass[] = "Cryptography";* //Enter your WIFI password

Memasukkan SSID dan Password yang tersedia untuk mendapatkan koneksi internet

```

void setup() {
  Serial.begin(9600);
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);

  Blynk.begin(auth, ssid, pass);
}

```

Pada Bagian ini pin yang sudah di definisikan diubah menjadi output dan untuk memulai proses koneksi.

```

BLYNK_WRITE(V0) {
  forward = param.asInt();
}

```

```

BLYNK_WRITE(V1) {
  backward = param.asInt();
}

```

```

BLYNK_WRITE(V2) {
  left = param.asInt();
}

```

```

BLYNK_WRITE(V3) {
  right = param.asInt();
}

```

```

BLYNK_WRITE(V4) {
    Speed = param.asInt();
}

```

Dan untuk di bagian ini untuk mengisi parameter untuk Virtual Port yang sudah ditentukan dan di konfigurasi pada aplikasi Blynk.

```

void mobilpintar() {
    if (forward == 1) {
        mobilmaju();
        Serial.println("mobilmaju");
    } else if (backward == 1) {
        mobilmundur();
        Serial.println("mobilmundur");
    } else if (left == 1) {
        mobilbelokkiri();
        Serial.println("mobilbelokkiri");
    } else if (right == 1) {
        mobilbelokkanan();
        Serial.println("mobilbelokkanan");
    } else if (forward == 0 && backward == 0 && left == 0 && right == 0) {
        mobilBerhenti();
        Serial.println("mobilBerhenti");
    }
}

```

Membuat fungsi ketika sketch sudah dijalankan atau perangkat sudah menyala, dan mendeklarasikan setiap fungsi yang akan diperintahkan berdasarkan pin mode yang sudah di definisikan.

```

}
void loop() {
  Blynk.run();
  mobilpintar();
}

```

Fungsi dari kode ini ketika fungsi setup sudah dijalankan maka fungsi ini bertujuan untuk mengeksekusi dan menjalankan program yang sudah dibuat berulang kali secara berkala.

```

void mobilmaju() {
  analogWrite(ENA, Speed);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

```

```

void mobilmundur() {
  analogWrite(ENA, Speed);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

```

```

void mobilbelokkiri() {
  analogWrite(ENA, Speed);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, HIGH);

```

```

digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
}

void mobilbelokkanan() {
  analogWrite(ENA, Speed);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void mobilBerhenti() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}

```

Pada bagian terakhir ini, melakukan fungsi pengoperasian pada mobil untuk berbelok sesuai dengan yang diarahkan.

#### 4.7 Tabel Pengujian

Arus listrik yang diberikan pada mobil :	Tegangan 12V (Voltase)	Arus Listrik A (Ampere)
OUT1	9.55	7.72
OUT2	9.43	7.64
OUT3	9.41	7.62
OUT4	9.35	7.59

Tabel 4.7 Tebel Arus Listrk yang Diberikan Pada Mobil

Arus listrik mobil saat tidak menyala :	Tegangan 12V (mV)
OUT1	03.7
OUT2	03.8
OUT3	02.3
OUT4	02.2

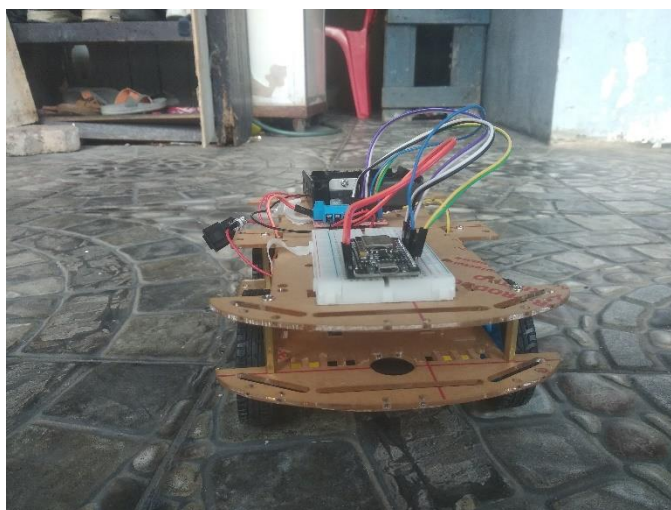
Tabel 4.7 Tebel Arus Listrik Mobil Tidak Menyala

Pergerakan Mobil saat :	TEGANGAN ARUS PADA MOTOR			
	OUT1 (V)	OUT2 (V)	OUT3 (V)	OUT4 (V)
Maju	9.33	9.33	9.33	9.33
Mundur	9.33	9.33	9.33	9.33
Belok Kanan	0.75	7.23	8.52	8.52
Belok Kiri	7.23	8.23	0.73	8.23

Tabel 4.7 Tebel Arus Listrik Ketika Mobil Berjalan

## 4.8 Gambar Alat

### 4.8.1 Tampak Depan



Gambar 4.8.1 Tampak Depan Mobil

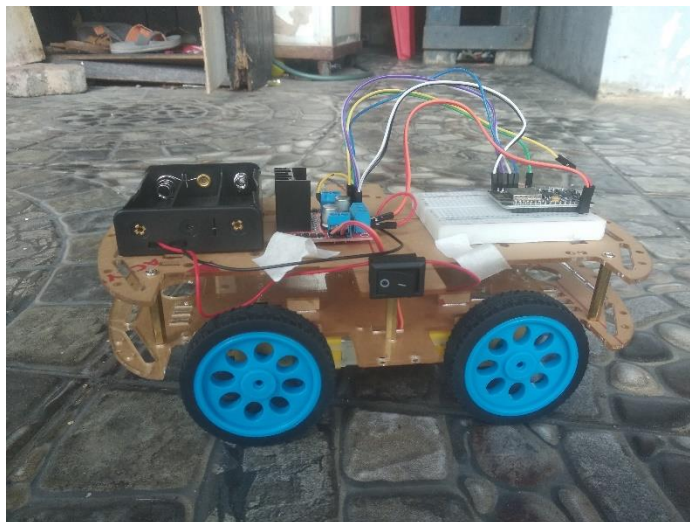


#### 4.8.2 Tampak Belakang



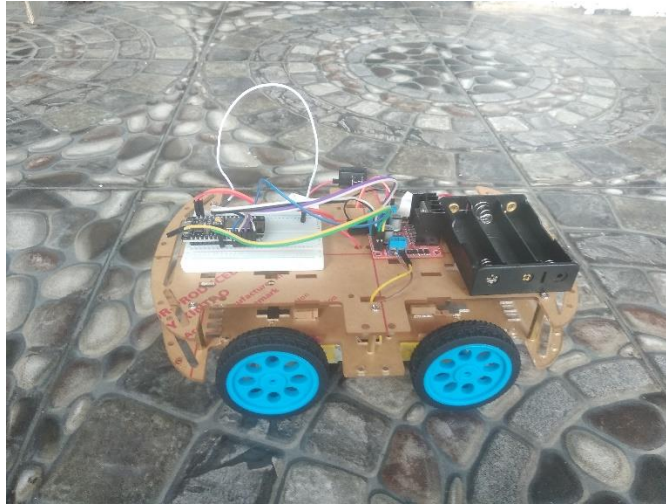
*Gambar 4.8.2 Tampak Belakang Mobil*

#### 4.8.3 Tampak Samping Kanan



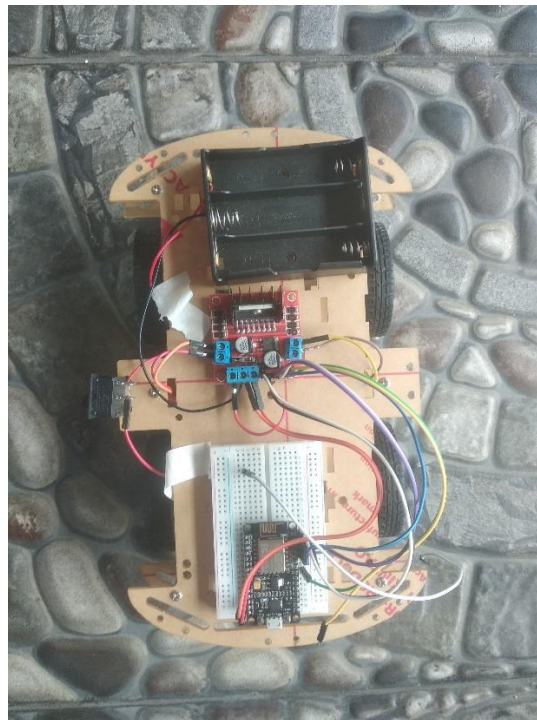
*Gambar 4.8.3 Tampak Samping Kanan*

#### 4.8.4 Tampak Samping Kiri



*Gambar 4.8.4 Tampak Samping Kiri*

#### 4.8.5 Tampak Atas



*Gambar 4.8.5 Tampak Atas*

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Prototype Delivery Robot menggunakan NodeMCU ESP8266 telah berhasil dirancang dan dijalankan, maka dapat diambil beberapa kesimpulan:

1. Prototype delivery robot menggunakan NodeMCU ESP8266 bisa dijalankan.
2. Prototype delivery robot menggunakan NodeMCU ESP8266 dapat memberikan perintah maju, mundur, belok kanan dan belok kiri.
3. Prototype delivery robot menggunakan NodeMCU ESP8266 ketika memberikan perintah maju dan mundur kepada Motor Driver L298N dan TT Motor DC akan bergerak dengan tegangan 9.33 V.
4. Prototype delivery robot menggunakan NodeMCU ESP8266 dapat melakukan belok kiri dan belok kanan dengan memberikan tegangan full 8.52 V disalah satu sisinya dan di satu sisi lainnya akan diberikan beban dengan memberikan sedikit tegangan untuk sebagai penompang berbelok dengan tegangan 0.73 V.

#### **5.2 Saran**

Alat yang ditulis dalam laporan ini dapat dikembangkan dengan menambahkan beberapa fitur untuk meningkatkan kemampuan dari alat prototype delivery robot ini adalah :

1. Penambahan alat Ultrasonik agar tidak terjadi tabrakan terhadap barang ataupun orang.
2. Diperlukannya Camera agar memudahkan pengantaran barang agar dapat dikontrol melalui sisi kontroller.

## DAFTAR PUSTAKA

- Al Khairi, M Habib. *Tutorial Lengkap Menggunakan Driver L298N dengan Arduino*. 18 September 2021. <https://www.mahirelektro.com/2020/02/tutorial-menggunakan-driver-motor-l298n-pada-Arduino.html> (diakses Januari 13, 2022).
- allgoblog.com. *Apa itu Arduino IDE dan Arduino Sketch ?* 26 Oktober 2017. <http://allgoblog.com/apa-itu-arduino-ide-dan-arduino-sketch/> (diakses Januari 13, 2022).
- arduino.cc. *Arduino IDE*. t.thn. <https://www.arduino.cc/en/software> (diakses Januari 13, 2022).
- erintafifah. *Mengenal Perangkat Lunak Arduino IDE*. 8 Oktober 2021. <https://www.kmtech.id/post/mengenal-perangkat-lunak-arduino-ide> (diakses Januari 13, 2022).
- Erwin Techable. *Mikrokontroler*. t.thn. <https://erwins.teachable.com/courses/775238/lectures/14057659> (diakses Februari 6, 2022).
- IMMERSA LAB. *SEJARAH MIKROKONTROLLER*. 16 Juni 2014. <https://www.immersa-lab.com/sejarah-mikrokontroler.htm> (diakses Februari 6, 2022).
- Nyebartilmu.com. *Mengenal Aplikasi BLYNK untuk fungsi IOT*. 23 November 2017. <https://www.nyebartilmu.com/mengenal-aplikasi-blynk-untuk-fungsi-iot/> (diakses Februari 23, 2022).
- Saputro, Teddy Tri. *Mengenal NodeMCU: Pertemuan Pertama*. 19 April 2007. <https://embeddednesia.com/v1/tutorial-nodemcu-pertemuan-pertama/> (diakses Januari 13, 2022).

Tekno Info Kamu. *Sejarah Mikrokontroller*. 1 Agustus 2021.

<https://www.teknoinfokamu.com/2021/07/sejarah-mikrokontroler.html>

(diakses Februari 6, 2022).

Ultra Fire. *Ultra Fire 18650 Battery*. 1 Januari 2018.

<https://www.ultrafire.com/ultra-fire-18650-battery-a79> (diakses Januari

13, 2022).