

PENERAPAN INTERNET OF THINGS (IOT) MENGGUNAKAN APLIKASI BLYNK PADA PURWARUPA LENGAN ROBOT SEBAGAI ROBOT PEMETIK CABAI

Alam Arthansyah
Fakultas Ilmu Komputer dan Teknologi Informasi
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
amarthansyah123@gmail.com

4 Juli 2020

Ringkasan

Perkembangan teknologi khususnya dunia Robotika yang pesat membawa dunia menuju sebuah zaman dimana pada beberapa aspek pekerjaan mulai digantikan dengan robot. Dalam berbagai kasus robot menjadi pilihan utama karena dapat mengurangi cost pengeluaran bagi perusahaan, selain itu hasil pekerjaan yang dilakukan oleh robot lebih presisi dan lebih detail dengan waktu penggerjaan yang lebih singkat dibandingkan dengan menggunakan tenaga manusia. Beberapa kasus kombinasi Manusia dan Robot menjadi solusi untuk permasalahan yang khusus yang tidak bisa dilakukan hanya dengan menggunakan robot saja. Berdasarkan keterangan di atas, kami bermaksud akan merancang sebuah sistem kontrol gabungan antara Blynk sebagai Dashboard pengendalinya dan mikrokontroler berbasis Arduino UNO untuk mengendalikan grabber robot yang dapat digunakan untuk memindahkan barang.

Keywords : Arduino, NodeMCU ESP8266, L298N Motor Driver, Micro Servo SG90, DC Motor.

1 PENDAHULUAN

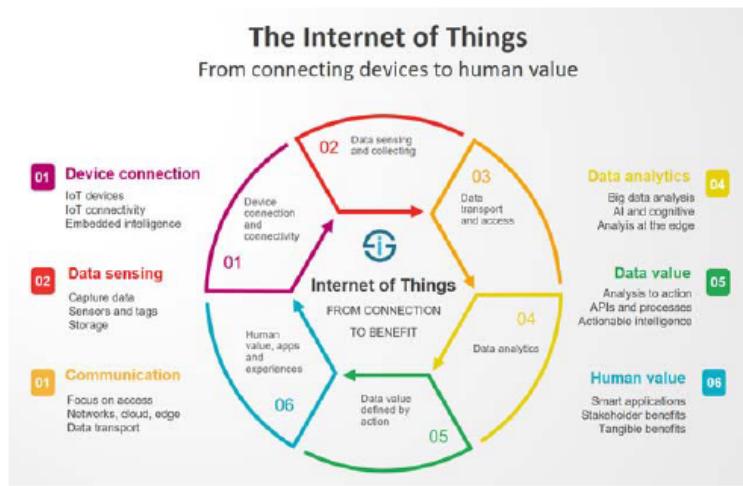
Perkembangan dunia teknologi khususnya dunia Robotika yang pesat membawa dunia menuju sebuah zaman dimana pada beberapa aspek pekerjaan mulai digantikan dengan robot. Dalam berbagai kasus robot menjadi pilihan utama karena dapat mengurangi cost pengeluaran bagi perusahaan selain itu hasil pekerjaan yang dilakukan oleh robot lebih presisi dan lebih detail dengan waktu pekerjaan yang lebih singkat dibandingkan dengan menggunakan tenaga manusia. Di beberapa kasus kombinasi Manusia dan Robot menjadi solusi untuk permasalahan yang khusus seperti dalam kasus penjinakkan bom, memindahkan barang – barang dengan beban yang berat di butuhkan control manusia untuk lebih fleksibel dalam menyelesaikan pekerjaan dimana akan sangat sulit jika hanya mengandalkan robot saja disini manusia mengendalikan robot untuk memindahkan barang di tempat yang belum ditentukan titik pemindahannya, sedangkan robot harus ditentukan dahulu titik pemindahan barang tersebut. Begitu juga ketika menjinakkan bom dimana setiap kasus memiliki rangkaian elektronika yang berbeda – beda.

Saat ini telah berkembang pula teknologi dari *smartphone* yang dapat digunakan untuk berbagai aplikasi. Penggunaan *smartphone* sebagai unit pengendali akan memberikan tingkat kepraktisan yang tinggi, mengingat piranti ini mudah digunakan dan telah menjadi bagian dari perlengkapan yang sehari-hari selalu dibawa oleh manusia.

Berdasarkan pertimbangan di atas, melalui penelitian ini akan dirancang sebuah sistem kontrol gabungan antara *Blynk* sebagai Dashboard pengendalinya dan NodeMCU berbasis IOT untuk mengendalikan *grabber robot* yang dapat digunakan untuk memindahkan barang.

2 Internet of Things

Internet of Things (IoT) merupakan kumpulan benda-benda (*things*), berupa perangkat fisik (*hardware/embedded system*) yang mampu bertukar informasi antar sumber informasi, operator layanan ataupun perangkat lainnya yang terhubung kedalam sistem sehingga dapat memberikan kemanfaatan yang lebih besar. Perangkat fisik (*hardware/embedded system*) dalam infrastruktur *Internet of Things* merupakan *hardware* yang tertanam (*embedded*) dengan elektronik, perangkat lunak, sensor dan juga koneksi.



Gambar 1: *Internet of Things*

2.1 Blynk

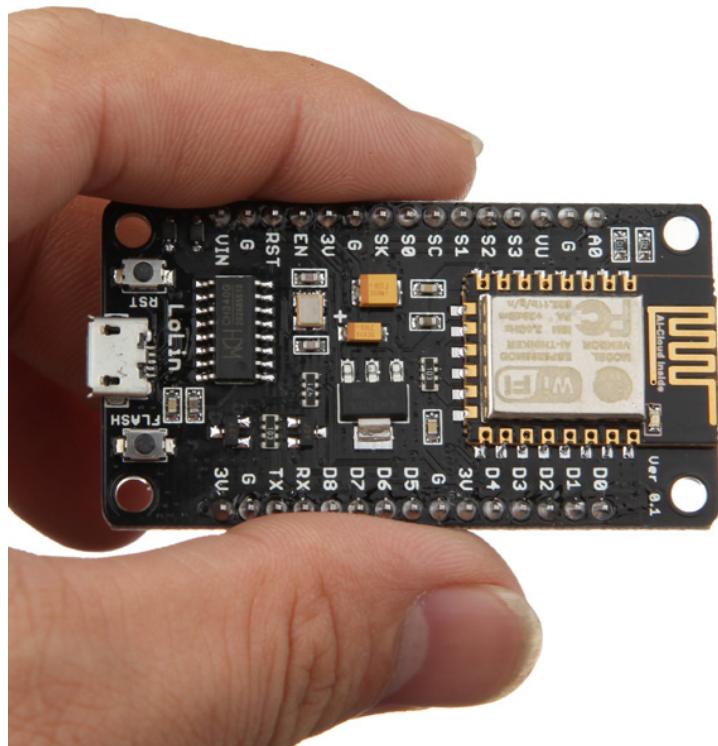


Gambar 2: blynk

Blynk adalah platform aplikasi yang dapat diunduh secara gratis untuk iOS dan Android yang berfungsi mengontrol Arduino, Raspberry Pi dan sejenisnya melalui Internet. Blynk dirancang untuk Internet of Things dengan tujuan dapat mengontrol hardware dari jarak jauh, dapat menampilkan data sensor, dapat menyimpan data, visual dan melakukan banyak hal canggih lainnya. Ada tiga komponen utama dalam platform yaitu Blynk App, Blynk Server, dan Blynk Library.

2.2 NodeMCU

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.



Gambar 3: NodeMCU

2.3 DC Motor

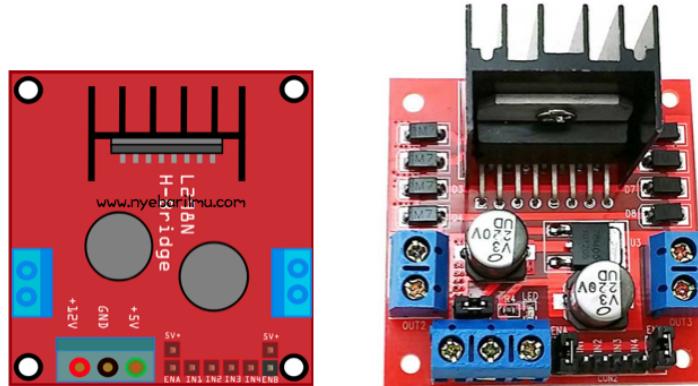
Motor DC adalah jenis motor listrik yang bekerja menggunakan sumber tegangan DC. Motor DC atau motor arus searah sebagaimana namanya, menggunakan arus langsung dan tidak langsung/direct-unidirectional. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan torque yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas.



Gambar 4: DC Motor

2.4 Motor Driver L298N

Driver motor L298N merupakan module driver motor DC yang paling banyak digunakan atau dipakai di dunia elektronika yang difungsikan untuk mengontrol kecepatan serta arah perputaran motor DC.



Gambar 5: Motor Driver L298N

IC L298 merupakan sebuah IC tipe H-bridge yang mampu mengendalikan beban-beban induktif seperti relay, solenoid, motor DC dan motor stepper.

Pada IC L298 terdiri dari transistor-transistor logik (TTL) dengan gerbang nand yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor dc maupun motor stepper.

Untuk dipasaran sudah terdapat modul driver motor menggunakan ic l298 ini, sehingga lebih praktis dalam penggunaannya karena pin I/O nya sudah terpackage dengan rapi dan mudah digunakan.

Kelebihan akan modul driver motor L298N ini yaitu dalam hal kepresision dalam mengontrol motor sehingga motor lebih mudah untuk dikontrol.

2.5 Motor Servo



Gambar 6: Motor Servo

Motor servo adalah sebuah motor dengan sistem closed feedback di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tam-pak pada gambar dengan pulsa 1.5 mS pada periode selebar 2 mS maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.

Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor stepper. Walau demikian, untuk beberapa keperluan tertentu, motor servo dapat dimodifikasi agar bergerak kontinyu.

Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagian bagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar.

Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan duty cycle sinyal PWM pada bagian pin kontrolnya. Motor Servo tampak pada gambar 4.

3 Perancangan perangkat keras

Cara kerja *Grabber Robot* yang akan dirancang ini adalah Sitem Kendali sebuah robot lengan oleh mikrokontroler NodeMCU dimana data mengenai pergerakannya dikirim dari aplikasi Blynk yang dikembangkan dengan bahasa pemrograman Arduino. Aplikasi *blynk* akan mengirim sekaligus menerima data internet ke mikrokontroler NodeMCU. Mikrokontroler NodeMCU akan memproses data ini untuk mengendalikan robot lengan melalui motor servo sebagai aktuatornya sehingga dapat bergerak sesuai dengan *slider* pengendali yang digeser pada aplikasi *blynk*.

Untuk memahami cara kerja sistem, maka dibuat blok sistem dari perancangan robot yang dibuat. Gambar 3.1 merupakan gambar blok sistem perancangan Pengendalian Robot Lengan 4 DOF menggunakan *blynk*. Sistem ini dibagi menjadi dua bagian yaitu Perancangan perangkat keras dan Perancangan perangkat lunak.



Gambar 7: Skema Kendali Arm Robot

Penjelasan diagram blok pada perancangan perangkat keras dari pengendalian Robot Lengan 4 DOF dengan aplikasi *blynk* berbasis IoT adalah sebagai berikut :

1. Bagian *Smartphone*

Merupakan perangkat yang digunakan untuk mengirim dan menerima data melalui aplikasi *Blynk*. Bagian ini membutuhkan koneksi internet agar dapat terhubung ke *blynk server*.

2. Bagian Mikrokontroler NodeMCU

Bagian ini berfungsi sebagai pusat kendali dari seluruh sistem yang ada, karena kode program utama terletak pada bagian ini. Merupakan bagian kontroler, semua data-data yang dikirim dari *blynk server* akan dibaca oleh kode program dan disimpan di dalam mikrokontroler NodeMCU. Mikrokontroler yang digunakan adalah mikrokontroler NodeMCU dengan kapasitas *flash memory* sebesar 16MB max (512K normal).

3. Bagian Robot Lengan

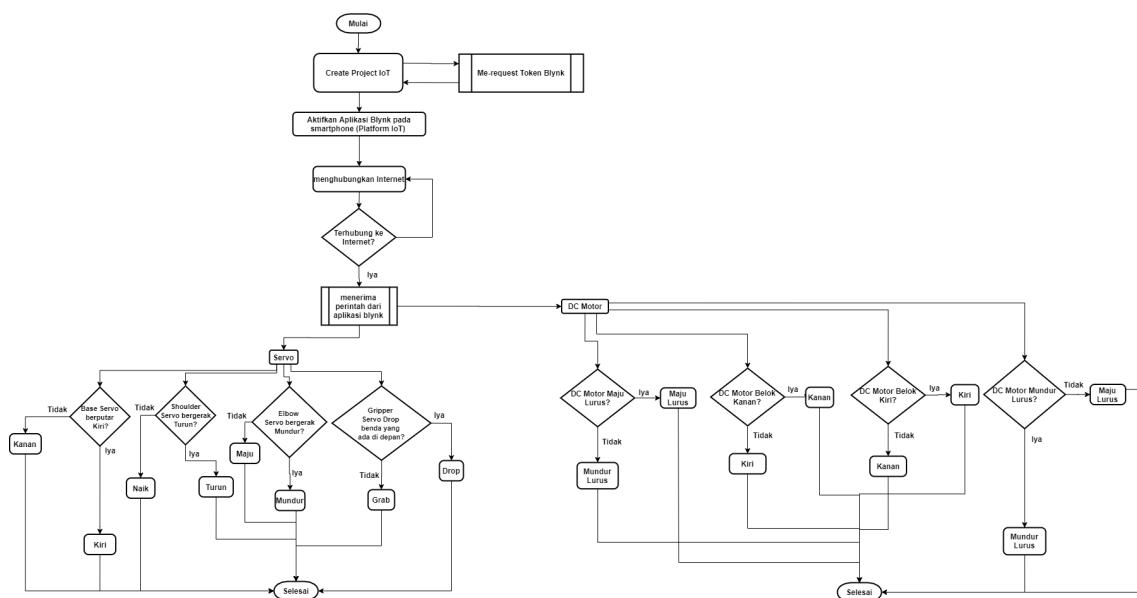
Merupakan *prototype manipulator* mekanik yang terdiri dari empat *joint* dan satu *end-effector*, dimana setiap *joint* dihubungkan oleh *link* yang dicetak dari bahan *acrylic*. Dimana aktuator robot lengan menggunakan motor servo sebagai penggerak sendi-sendi robot yang berada di setiap *joint* dan *end-effector*.

4. Bagian Motor DC

Bagian ini berfungsi sebagai penggerak *Grabber Robot*. Motor DC ini juga dapat disebut sebagai Motor Arus Searah. Seperti namanya, DC Motor memiliki dua terminal dan memerlukan tegangan arus searah atau DC (Direct Current) untuk dapat menggerakannya.

3.1 Perancangan perangkat lunak

Pembuatan program menggunakan bahasa Arduino. Program dirancang agar mikrokontroler dapat mengirim dan menerima data secara serial sehingga robot lengan dapat bergerak sesuai perintah yang di instruksikan melalui *Blynk*.



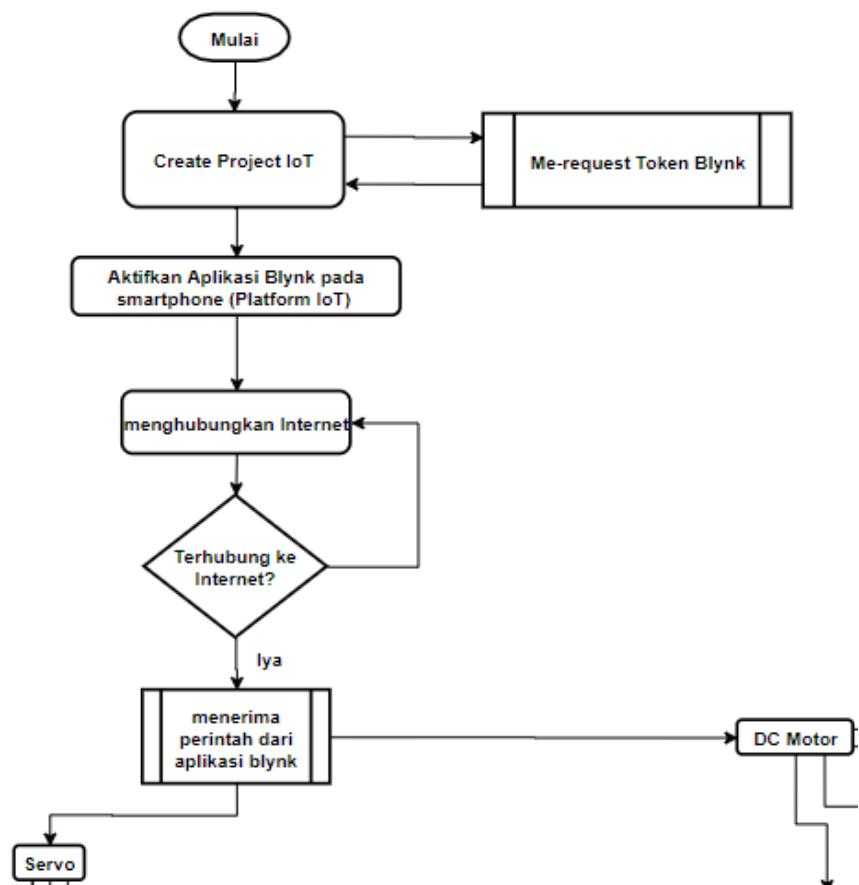
Gambar 8: Perancangan perangkat lunak

4 Analisa Flowchart Secara Detail

Untuk analisa flowchart ini dibagi menjadi 3 bagian diantaranya:

1. Proses pengkoneksian blynk dengan jaringan Wi-Fi pada *smartphone*.

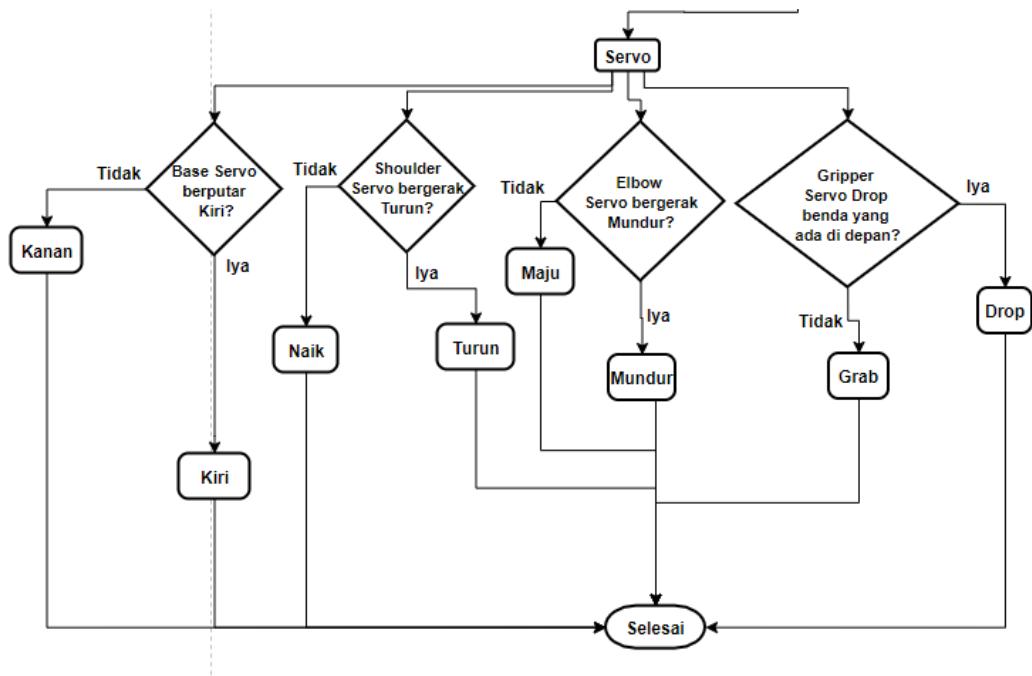
Sebelum membuat sebuah *remote control* pada aplikasi *blynk*, langkah awal ialah dengan *Login/Create new account/Scan QR Code*. jika belum memiliki sebuah akun pada aplikasi *blynk* maka bisa membuat akun baru atau dapat *Login* dengan menggunakan akun *Facebook* jika memiliki akun *Facebook* atau dapat *Scan QR Code* untuk *sharing remote control* tanpa perlu *Login* menggunakan 2 akun, kemudian membuat *remote control* pada aplikasi *blynk*, setelah membuat lakukan pengiriman *token blynk* agar *blynk* dan mikrokontroler dapat dikendalikan, aktifkan aplikasi *blynk* pada *smartphone (platform IoT)*, lalu *blynk* akan menghubungkan ke *internet*, jika tidak terhubung maka lakukan koneksi ulang agar dapat terhubung ke *internet*, jika berhasil terhubung ke *internet* maka *remote control* pada aplikasi *blynk* dapat dikendalikan.



Gambar 9: Proses pengkoneksian blynk dengan jaringan Wi-Fi pada *smartphone*.

2. Proses penggerakan servo.

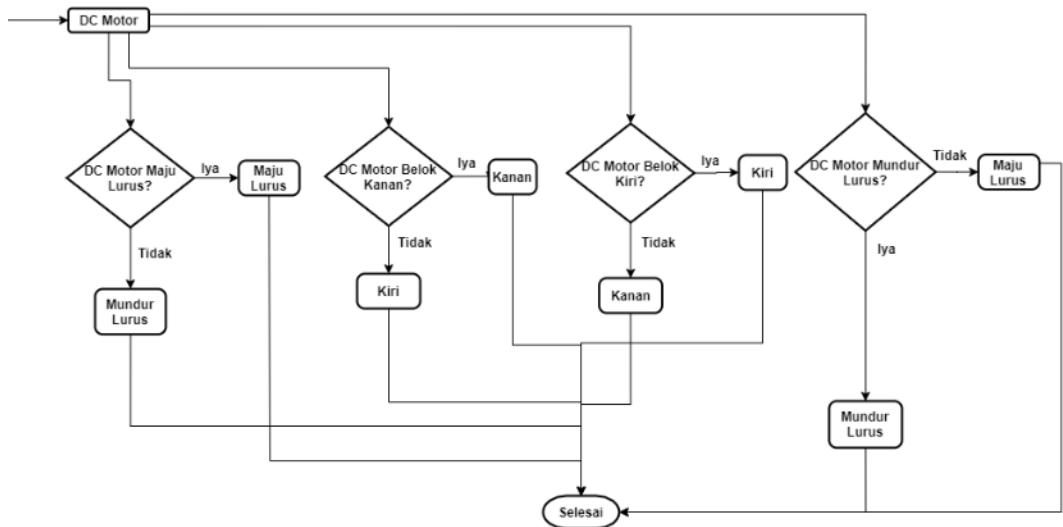
- Pada servo menerima perintah dari NodeMCU untuk berputar ke kiri, maka *base servo* akan berputar ke kiri, jika tidak maka *base servo* akan berputar ke arah kanan.
- Pada servo menerima perintah dari NodeMCU untuk bergerak turun, maka *shoulder servo* akan bergerak turun, jika tidak maka *shoulder servo* akan bergerak naik.
- Pada servo menerima perintah dari NodeMCU untuk bergerak mundur, maka *elbow servo* akan bergerak mundur, jika tidak maka *elbow servo* akan bergerak maju.
- Pada servo menerima perintah dari NodeMCU untuk drop benda yang ada di depan, maka *gripper servo* akan drop benda yang ada di depannya, jika tidak maka *gripper servo* akan **grab** benda yang ada di depannya.



Gambar 10: Proses penggerakan motor servo

3. Proses penggerakan dc motor.

- Pada motor dc menerima perintah dari NodeMCU untuk bergerak maju lurus, jika iya maka motor dc akan bergerak maju lurus, jika tidak maka motor dc mundur lurus.
- Pada motor dc menerima perintah dari NodeMCU untuk belok kanan, jika iya maka motor dc akan belok kanan, jika tidak maka motor dc akan belok kiri.
- Pada motor dc menerima perintah dari NodeMCU untuk belok kiri, jika iya maka motor dc akan belok kiri, jika tidak maka motor dc akan belok kanan.
- Pada motor dc menerima perintah dari NodeMCU untuk mundur lurus, jika iya maka motor dc akan mundur lurus, jika tidak maka motor dc maju lurus.



Gambar 11: Proses penggerakan dc motor.

5 Perancangan Mekanik Lengan Robot



Gambar 12: Desain Mekanik Lengan Robot



Gambar 13: Prototipe Lengan Robot

Lengan robot dirancang dengan 4 DOF (*Degree Of Freedom*) ang terdiri dari 4 buah servo dengan konfigurasi sebagai berikut:

1. DOF 1 (Servo1):*Base Servo*
2. DOF 2 (Servo2):*Elbow Servo*
3. DOF 3 (Servo3):*Shoulder Servo*
4. DOF 4 (Servo4):*Gripper Servo*

Arm Robot ini memiliki spesifikasi mekanik dengan ketinggian 15cm, panjang lengan robot 25cm, lebar alas 14.1cm, panjang alas 9.2cm. *Base Servo* digunakan untuk memutar sebesar 180 derajat. *Elbow Servo* digunakan untuk menaikkan atau menurunkan. *Shoulder Servo* digunakan untuk maju dan mundur. *Gripper Servo* digunakan untuk mengambil dan melepas barang yang ada di depannya.

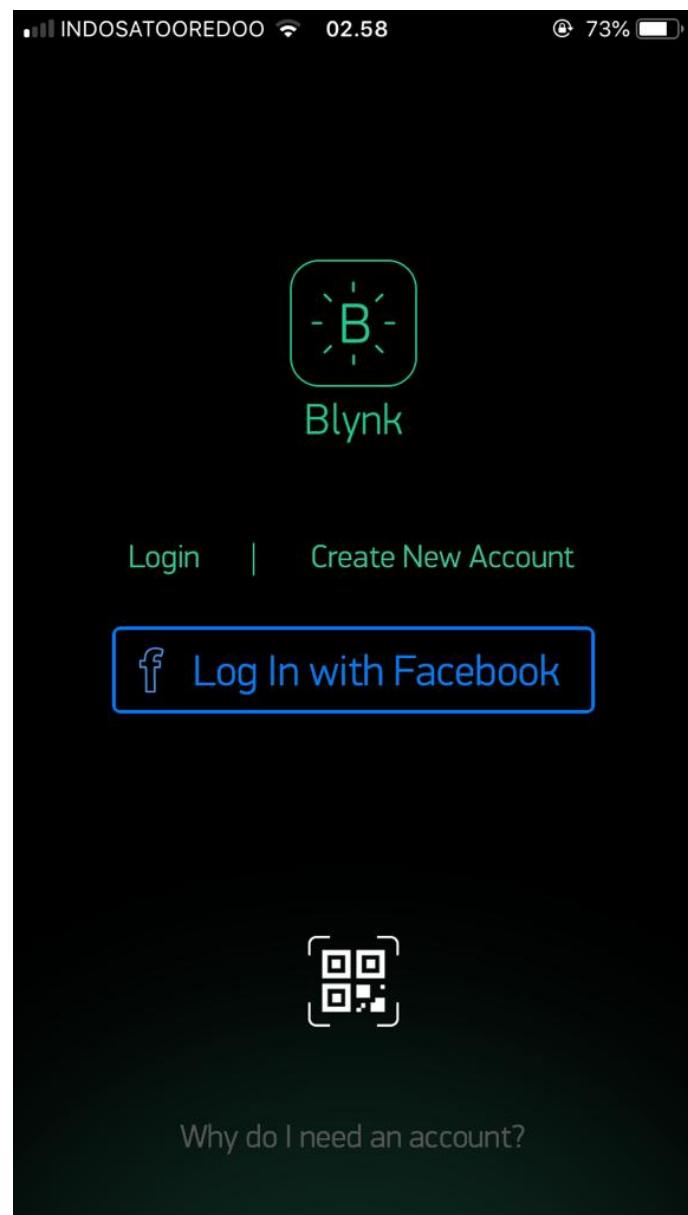
5.1 Pembuatan Remote Control pada Blynk

1. Download aplikasi blynk pada Playstore/AppStore
2. *Install blynk library.*

Jika memiliki sebuah NodeMcu atau mikrokontroler yang dapat terhubung ke internet maka untuk menggerakan programnya dibutuhkan sebuah library. Hal ini diharuskan karena untuk memprogram pada sebuah arduino IDE diwajibkan memiliki library, agar program dapat berjalan.

3. Tampilan awal blynk

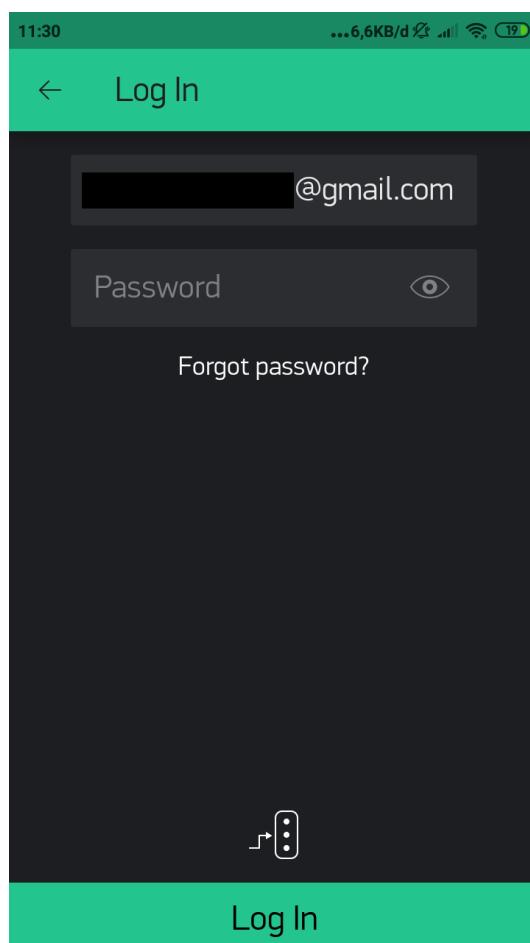
Sebelum membuat sebuah remote control pada aplikasi Blynk, langkah awal ialah dengan *Login/Create New Account/Scan QR Code*. Jika belum memiliki sebuah akun pada aplikasi Blynk maka bisa membuat akun baru atau dapat LogIn dengan menggunakan akun Facebook jika memiliki akun Facebook atau dapat *Scan QR Code* untuk *Sharing Remote Control* tanpa perlu *LogIn* menggunakan 2 akun.



Gambar 14: Tampilan awal blynk

4. Tampilan Log In

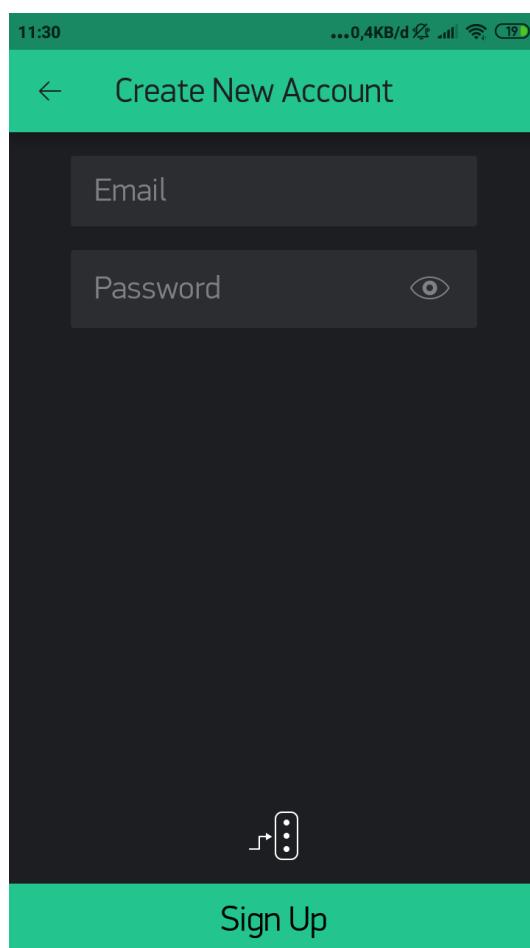
Jika sudah memiliki akun pada blynk dan tidak ingin LogIn dengan menggunakan facebook, maka dapat LogIn dengan menggunakan akun blynk.



Gambar 15: Tampilan Log In

5. Tampilan membuat akun baru

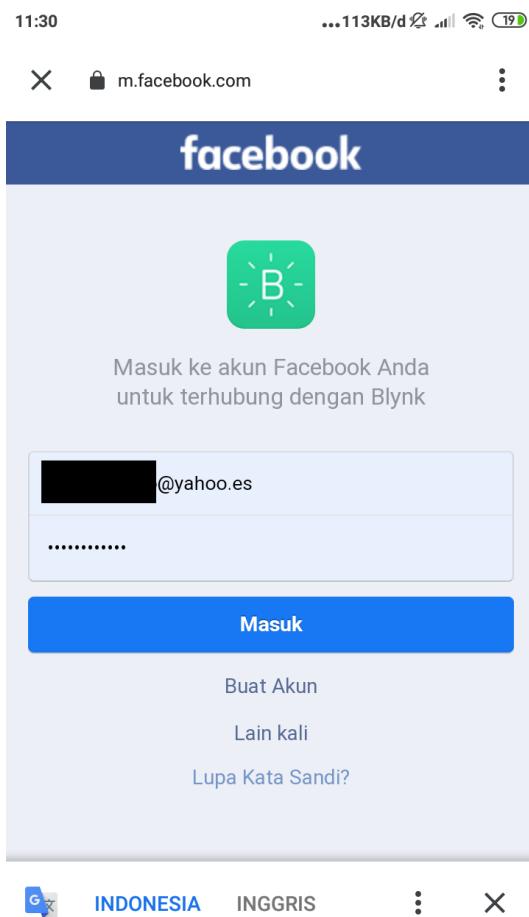
Tampilan ini adalah untuk membuat akun baru. Untuk membuat akun baru pada *blynk*, masukkan email dan password yang diinginkan, setelah itu pilih *Sign Up*, setelah *Sign Up* periksa email untuk mendapatkan bahwa akun telah terdaftar dan mendapatkan kode autentikasi.



Gambar 16: Tampilan membuat akun baru

6. Tampilan untuk LogIn dengan menggunakan Facebook

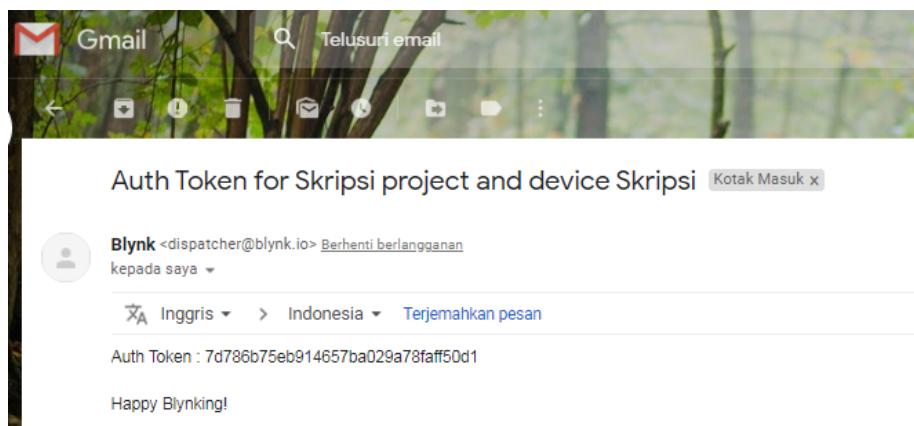
Jika tidak ingin membuat sebuah akun pada blynk, maka dapat *LogIn* dengan menggunakan akun Facebook. Setelah *LogIn* periksa email untuk mendapatkan bahwa akun telah terdaftar dan mendapatkan kode autentikasi.



Gambar 17: Tampilan Log In menggunakan Facebook

7. Tampilan kode autentikasi pada halaman gmail

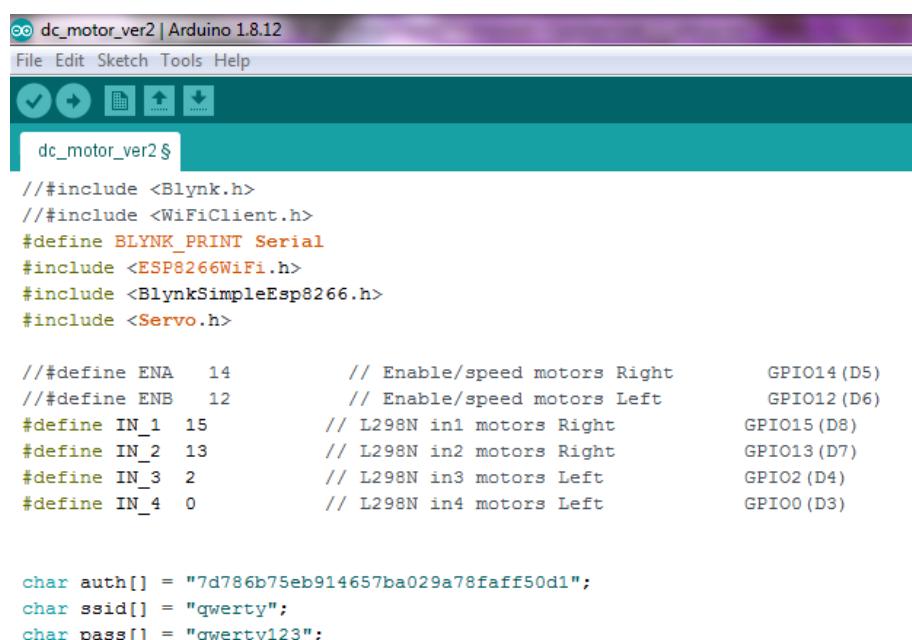
Jika sudah membuat sebuah akun pada blynk atau *LogIn* dengan menggunakan Facebook, maka akan dapat kode autentikasi pada gmail yang digunakan. Kode ini berfungsi sebagai penghubung antara blynk server dengan mikrokontroler yang digunakan. Setelah mendapatkan kode autentikasi maka *copy-paste* ke dalam arduino IDE.



Gambar 18: Tampilan kode autentikasi pada halaman gmail

8. Tampilan arduino IDE

Jika sudah mendapatkan kode autentikasi pada email yang telah di daftarkan pada akun blynk, maka copy-paste pada arduino IDE. Untuk memprogram sebuah akun blynk yang berasis IoT (*Internet of Things*), ada hal yang wajib di *input* ke dalam *sketch* arduino IDE, yaitu berupa *library blynk* (`# define BLYNK_PRINT Serial`), (`# include (BlynkSimpleEsp8266.h)`), dan *char auth, ssid, pass*. Tanpa *library* dan *char* yang telah disebutkan, maka blynk tidak dapat terhubung ke mikrokontroler.



```
dc_motor_ver2 | Arduino 1.8.12
File Edit Sketch Tools Help
dc_motor_ver2
//#include <Blynk.h>
//#include <WiFiClient.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>

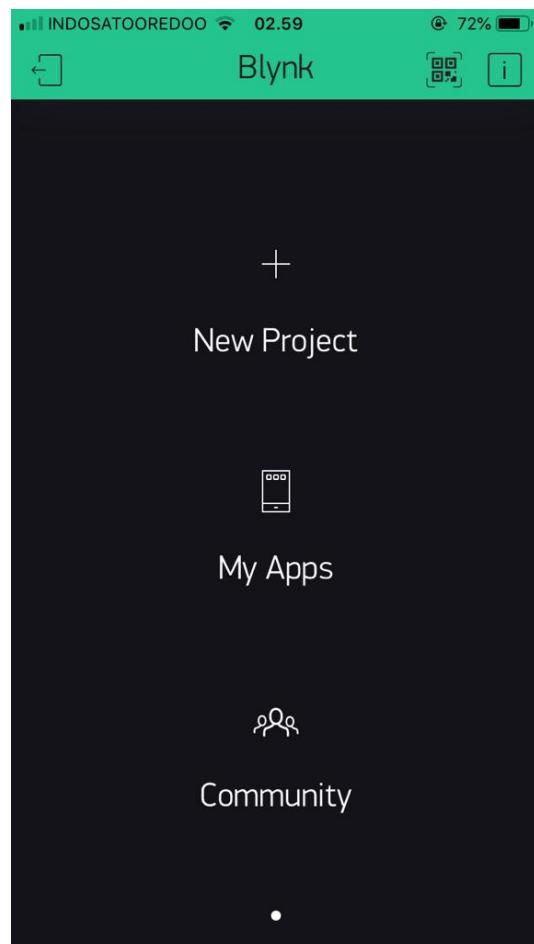
#define ENA 14      // Enable/speed motors Right      GPIO14 (D5)
#define ENB 12      // Enable/speed motors Left       GPIO12 (D6)
#define IN_1 15     // L298N in1 motors Right      GPIO15 (D8)
#define IN_2 13     // L298N in2 motors Right      GPIO13 (D7)
#define IN_3 2      // L298N in3 motors Left       GPIO2 (D4)
#define IN_4 0      // L298N in4 motors Left       GPIO0 (D3)

char auth[] = "7d786b75eb914657ba029a78faff50d1";
char ssid[] = "qwerty";
char pass[] = "qwerty123";
```

Gambar 19: Tampilan Arduino IDE

9. Tampilan kedua blynk

Pada tampilan kedua terdapat *New Project* (untuk membuat project baru), *My Apps*(jika sudah memiliki project), *Community* (sebuah komunitas atau sebuah forum untuk tanya jawab).

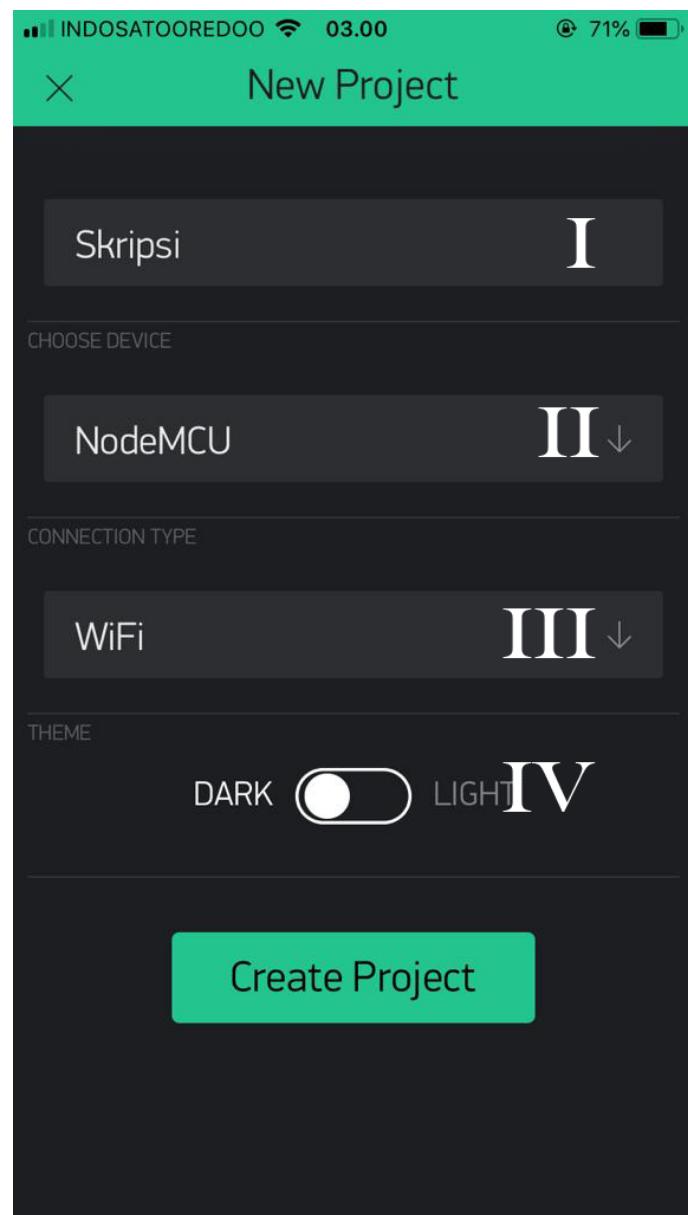


Gambar 20: Tampilan kedua blynk

10. Tampilan Membuat *Project* Baru.

Untuk membuat sebuah *project* pada aplikasi Blynk ialah dengan cara:

- I Masukkan nama *project* yang diinginkan
- II Pilih Jenis *device* atau jenis mikrokontroler yang digunakan.
- III Setelah itu pilih *Connection Type*, disini apakah menggunakan *Internet*, *Bluetooth*, *GSM*, *WiFi* atau *USB*. Untuk lebih jelasnya lihat pada gambar 3.16.
- IV Pilih jenis Tampilan pada aplikasi blynk apakah ingin menggunakan gelap atau terang.



Gambar 21: Membuat project baru

11. Memilih jenis koneksi

Untuk pada project ini, mikrokontroler yang di gunakan yaitu menggunakan NodeMCU, dan jenis koneksi yang di sediakan oleh aplikasi blynk berupa:

- **GSM**

GSM merupakan singkatan dari *Global System for Mobile Communications*. Jaringan *GSM* bisa diartikan sebagai sebuah teknologi komunikasi seluler yang bersifat *digital*. Teknologi *GSM* banyak diterapkan pada komunikasi bergerak, khususnya telepon genggam.

- **USB**

USB adalah singkatan dari *Universal Serial Bus* dan merupakan media penghubung antara komputer dengan perangkat-perangkat elektronik lainnya seperti mouse, keyboard, printer, scanner, ponsel, *flash drive*, *DVD writer*, Konsol permainan, kamera, modem dan bahkan digunakan sebagai media penghubung untuk mengendalikan alat-alat uji dan mesin-mesin produksi.

- **Wi-Fi**

”*Wireless Fidelity*” atau disingkat Wi-Fi adalah suatu teknologi yang memakai gelombang radio untuk menghubungkan perangkat (PC, laptop, smartphone) ke jaringan komputer. Atau definisi Wi-Fi yaitu teknologi yang menggunakan gelombang radio supaya komputer bisa mengakses internet.

- **Bluetooth**

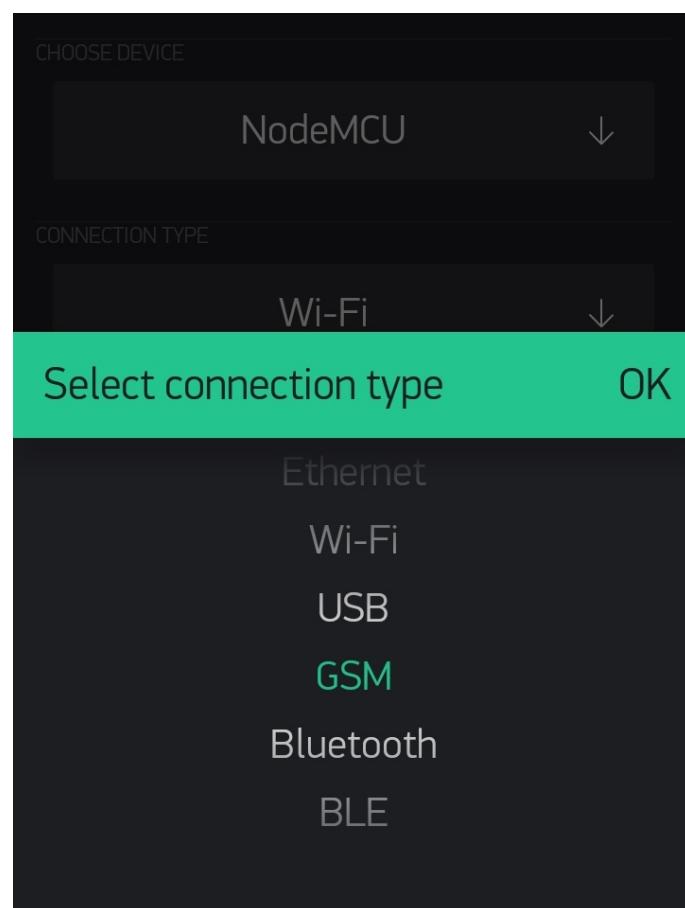
Bluetooth merupakan *chip radio* yang dimasukkan ke dalam komputer, printer, handphone dan sebagainya. *Chip bluetooth* ini dirancang untuk menggantikan kabel.

- **BLE**

Bluetooth Low Energy(BLE) merupakan perkembangan dari *Classic Bluetooth* yang khusus di implementasikan pada teknologi *Internet of Things* (IoT).

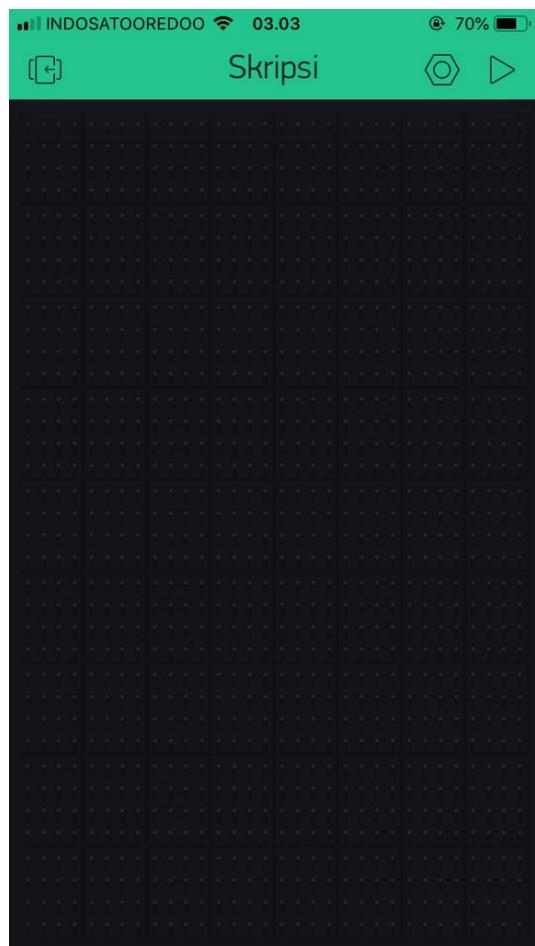
- *Ethernet*

Ethernet merupakan teknologi jaringan yang paling terkenal dan paling banyak digunakan di dunia saat ini. Dengan *Ethernet* kita bisa menghubungkan banyak komputer dengan biaya rendah dan fleksibilitas tinggi.



Gambar 22: Jenis koneksi

12. lembar project

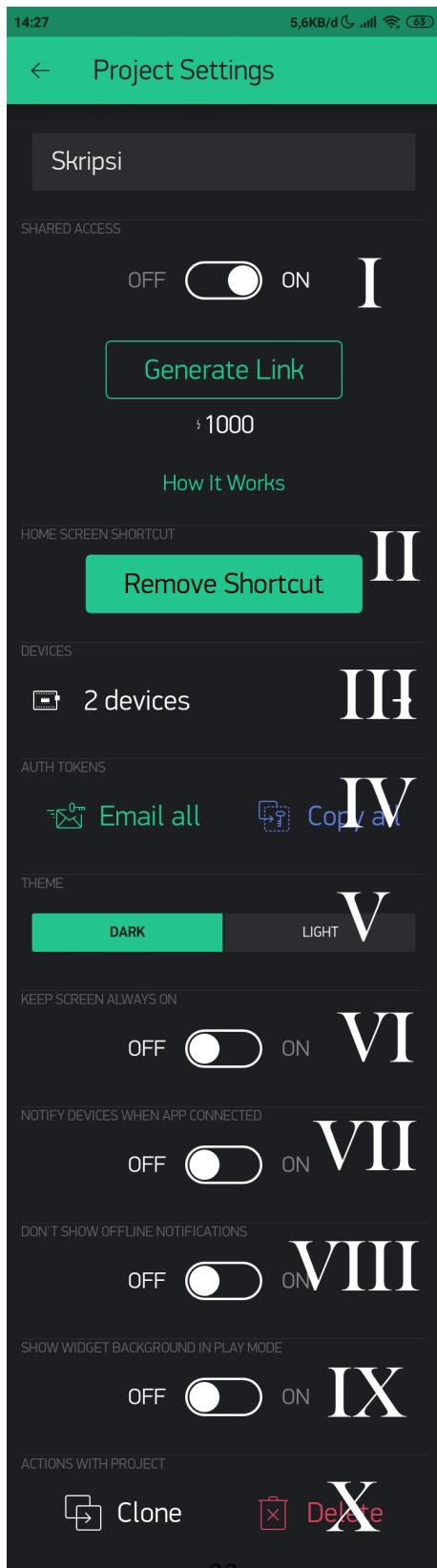


Gambar 23: lembar *project*

13. *Project Settings*

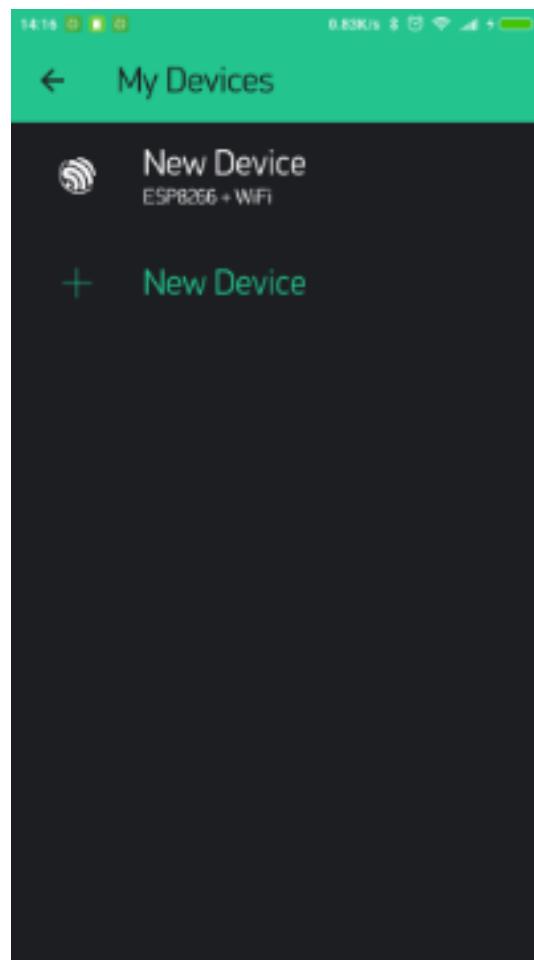
Pada bagian ini digunakan untuk mengatur *project* dan *sharing access remote control* dimana *project* yang dibuat tidak dapat dimodifikasi selain admin yang membuat *project* tersebut. Berikut penjelasan beberapa bagian yang terdapat pada *Project Settings*:

- I *Sharing access remote control* hanya bisa dilihat dan digunakan oleh *user* tetapi *user* tidak mendapatkan hak akses untuk mengatur dan merubah *project* tersebut, dengan catatan *project* yang di *shared* harus memiliki 1000 *energy* seperti pada gambar 3.21.
- II Menambahkan *shortcut* pada *home screen*.
- III Dapat menambahkan *device* baru dengan cara *click Devices* lalu pilih *New Devices*. Untuk lebih jelasnya lihat pada gambar 3.18.
- IV Untuk dapat mengendalikan sebuah *remote control* pada blynk yang telah dibuat, maka harus mengirim sebuah *AUTHENTICATION TOKEN* ke sebuah email yang telah digunakan pada akun blynk ini.
- V Mengubah tampilan menjadi gelap atau terang.
- VI *Keep screen always on*.
- VII Memberi notifikasi pada *device*, jika sudah terkoneksi dengan Wi-Fi.
- VIII Tidak memberikan notifikasi, jika blynk *offline*.
- IX Menampilkan pada *background*, jika blynk sedang berjalan.
- X Dapat *sharing project* tanpa perlu melalui admin, tetapi dengan syarat admin sudah memberikan hak akses kepada *user* agar *user* mendapatkan *cloning* dari admin, dengan cara *click* pada bagian *Clone* lalu *user scan QR Code* pada *smartphone* admin.

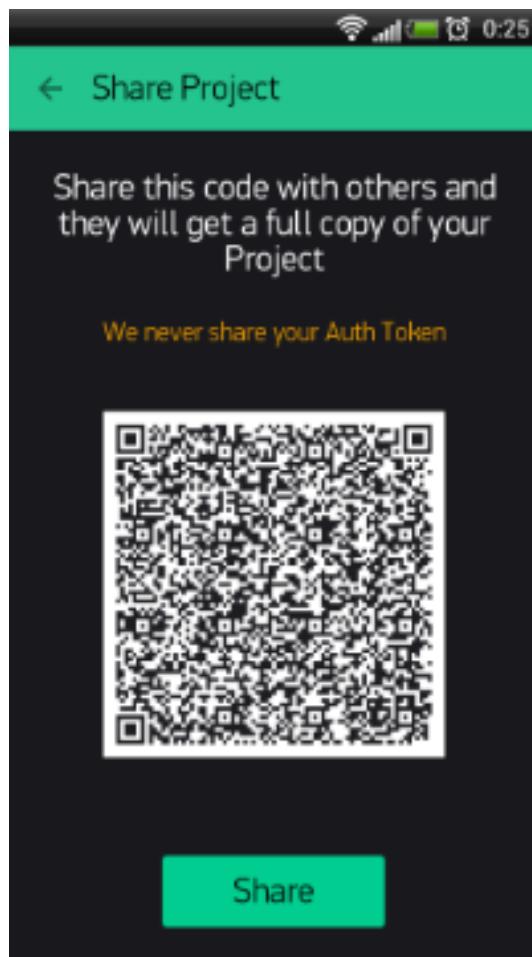


14. Menambah *devices* baru

Pada bagian ini, *device* baru dapat di tambahkan sebanyak yang di inginkan, dan terdapat berbagai jenis mikrokontroler yang dapat di pilih seperti, NodeMCU, Arduino Uno, dll..



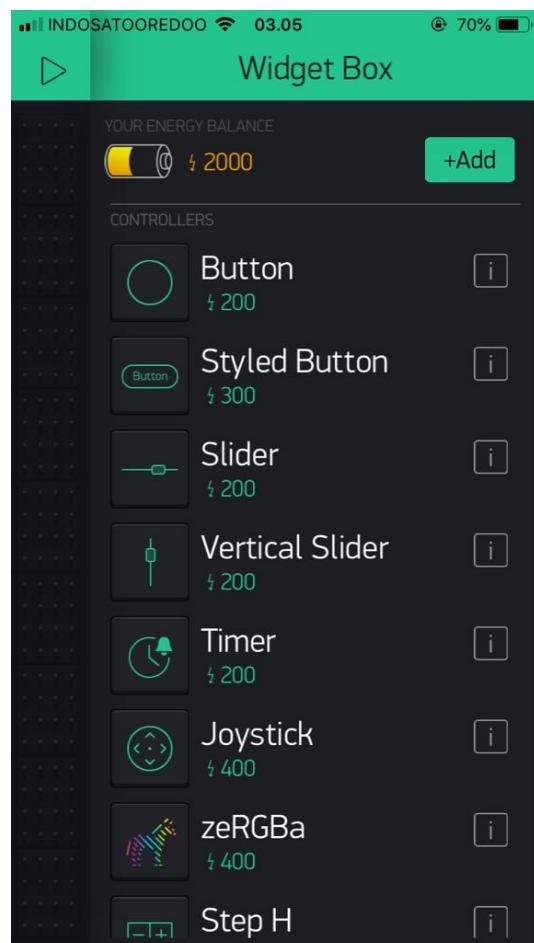
Gambar 25: Menambah *devices* baru



Gambar 26: *Share Project*

15. Widget Box

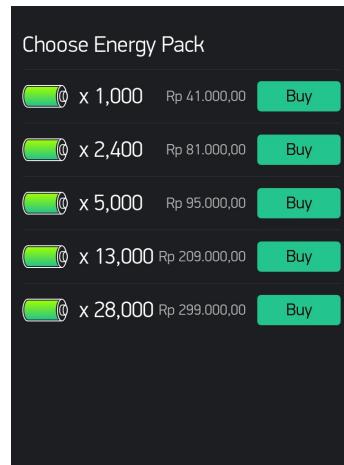
Di bagian *Widget Box* ini terdapat banyak *tools* yang digunakan untuk membuat sebuah *project* yang berbasis IoT.



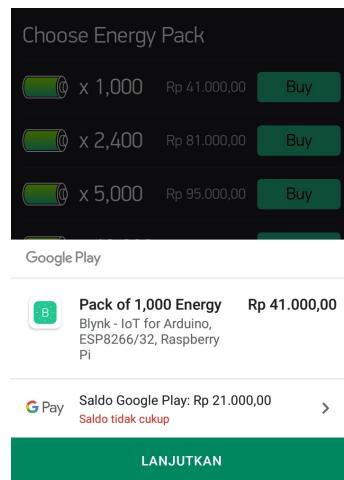
Gambar 27: *Widget Box*

16. Energy pada widget box

Pada bagian ini *energy* bawaan blynk sejumlah 2.000, namun untuk menambah *energy* hingga lebih banyak, di haruskan melakukan pembelian dan pembayaran dengan menggunakan *google play/iphone store*.



Gambar 28: Memilih *energy* untuk di beli

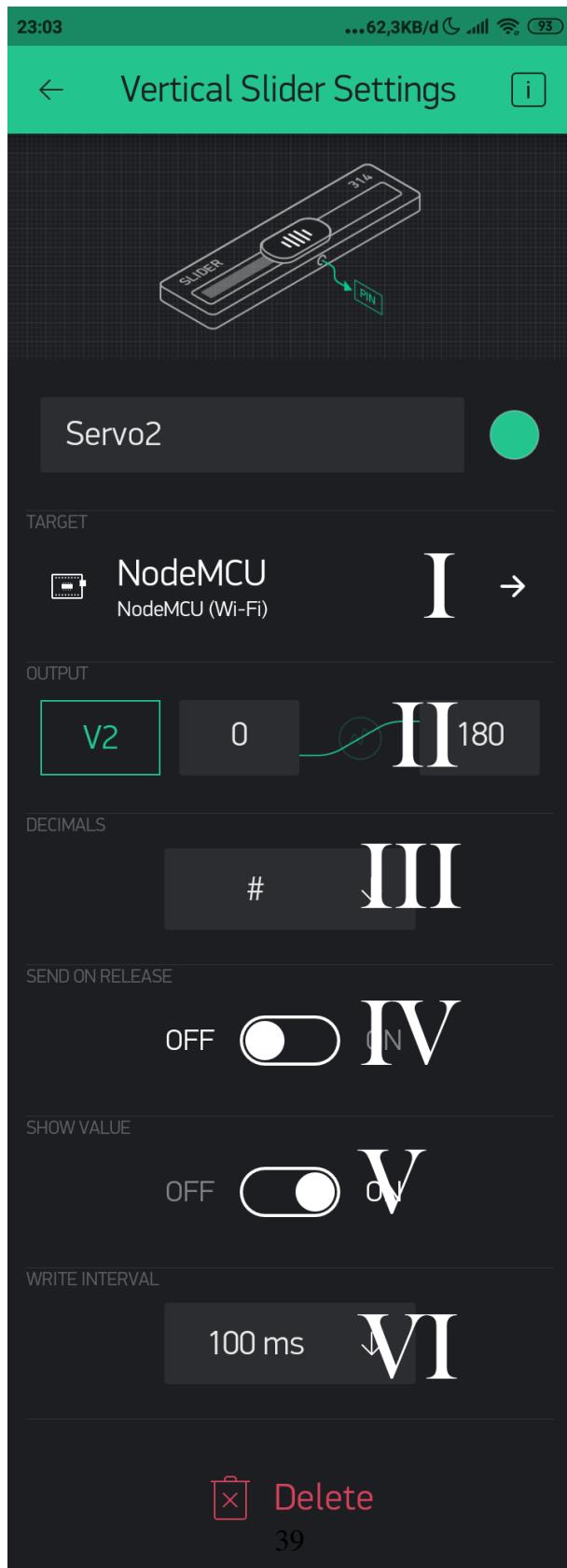


Gambar 29: Melakukan pembayaran menggunakan *google play*

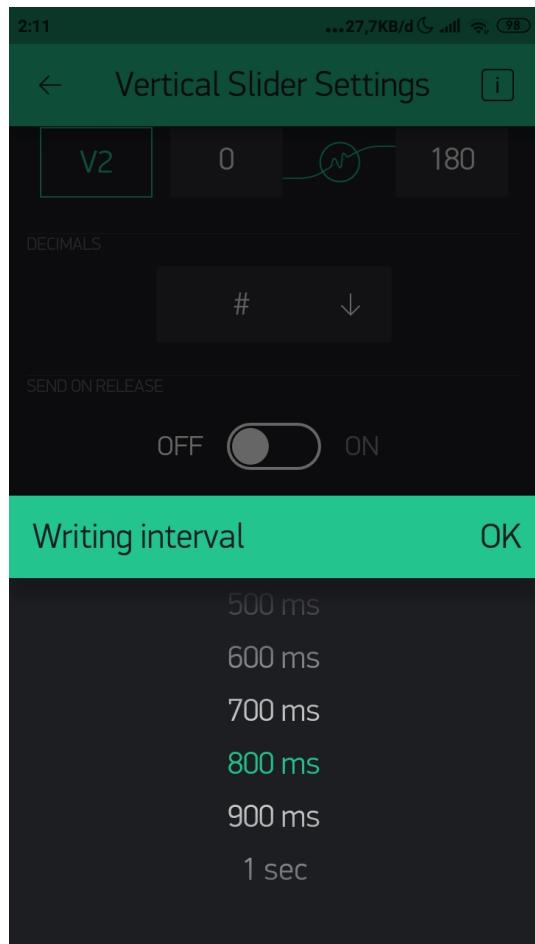
17. *Vertical Slider Settings*

Pada tampilan ini merupakan tampilan untuk mengatur slider pada *blynk remote control*. Fungsinya sama seperti *horizontal slider*. Perbedaannya ialah pada posisi slidernya. Kalau untuk *slider* ini posisinya dalam keadaan *vertical*, sedangkan *horizontal slider* dalam keadaan posisi *horizontal*. Slider ini di fungsikan untuk menggerakan jenis *Shoulder servo* dan *Elbow servo*. Berikut beberapa bagian yang terdapat pada bagian *slider settings*:

- I Jenis mikrokontroler yang di gunakan.
- II Jenis pin yang di gunakan, yaitu virtual pin.
- III *Decimals* ini di gunakan jika nilai outputnya bernali *decimals*.
- IV Pada bagian ini, jika di aktifkan, maka perubahan *output* tidak langsung terjadi saat slider digerakkan akan tetapi, terjadi saat jari melepas perubahan slider. Namun, jika tidak di aktifkan maka perubahan *output* akan langsung terjadi saat slider digerakkan.
- V Pada bagian ini, jika di aktifkan maka tampilan *vertical slider* akan menampilkan nilai derajat servo, namun jika tidak di aktifkan, maka nilai derajat servo tidak akan tampil di *vertical slider*.
- VI Pada bagian ini, di gunakan untuk mengirimkan sinyal kecepatan dari blynk menuju servo. Semakin kecil nilai *writing interval* (100 ms) maka akan semakin cepat proses pengiriman sinyal dari blynk ke servo, namun jika semakin besar nilai *writing interval* maka akan semakin lambat dalam proses pengiriman sinyal dari blynk ke servo. Untuk lebih jelasnya bisa di lihat pada gambar 3.25.



Gambar 30: *Vertical slider settings*

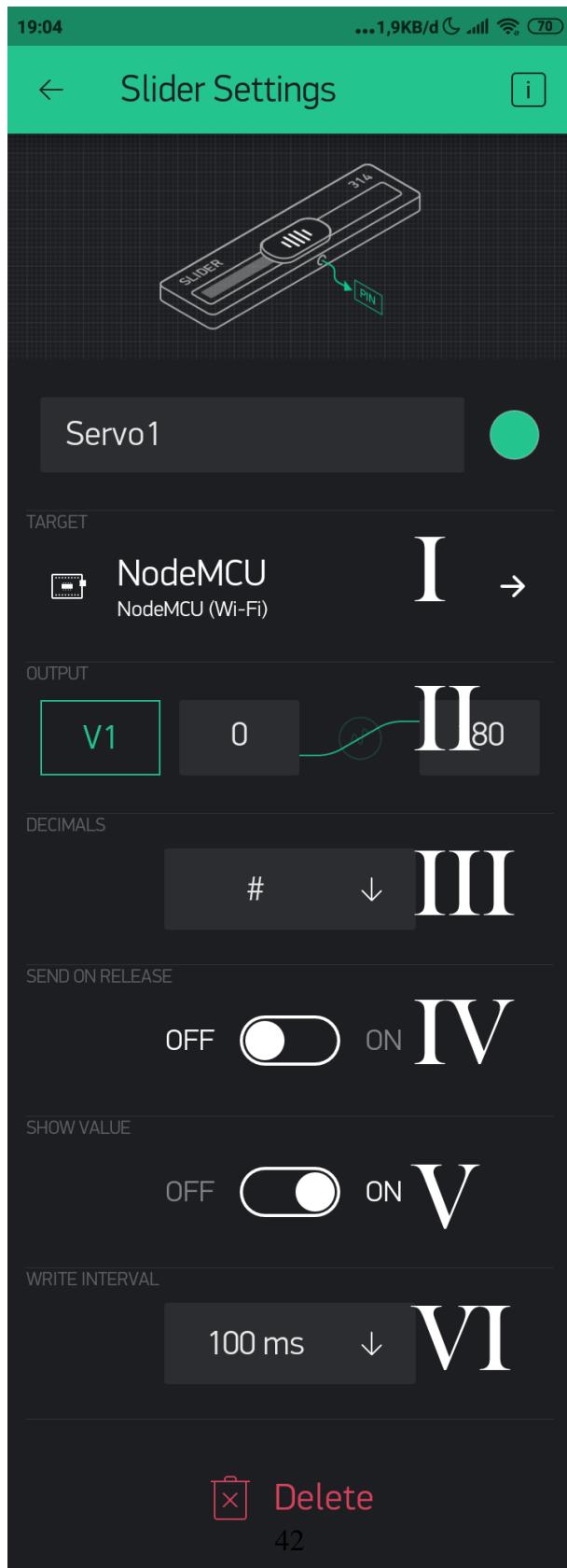


Gambar 31: *Writing interval*

18. *Slider Settings*

Pada tampilan ini merupakan tampilan untuk mengatur *slider* pada *blynk remote control*. Fungsinya sama seperti *Vertical Slider*, perbedaannya ialah pada posisi slidernya. Kalau untuk *slider* ini posisinya dalam keadaan *horizontal*, sedangkan *vertical slider* dalam keadaan posisi *vertical*. Slider ini di fungsikan untuk menggerakan jenis *base servo*. berikut beberapa bagian yang terdapat pada *slider settings*:

- I Jenis mikrokontroler yang di gunakan.
- II Jenis pin yang di gunakan, yaitu virtual pin.
- III *Decimals* ini di gunakan jika nilai outputnya bernilai *decimals*.
- IV Pada bagian ini, jika di aktifkan, maka perubahan *output* tidak langsung terjadi saat slider digerakkan akan tetapi, terjadi saat jari melepas perubahan slider. Namun, jika tidak di aktifkan maka perubahan *output* akan langsung terjadi saat slider digerakkan.
- V Pada bagian ini, jika di aktifkan maka tampilan *vertical slider* akan menampilkan nilai derajat servo, namun jika tidak di aktifkan, maka nilai derajat servo tidak akan tampil di *vertical slider*.
- VI Pada bagian ini, di gunakan untuk mengirimkan sinyal kecepatan dari blynk menuju servo. Semakin kecil nilai *writing interval* (100 ms) maka akan semakin cepat proses pengiriman sinyal dari blynk ke servo, namun jika semakin besar nilai *writing interval* maka akan semakin lambat dalam proses pengiriman sinyal dari blynk ke servo. Untuk lebih jelasnya bisa di lihat pada gambar 3.25.

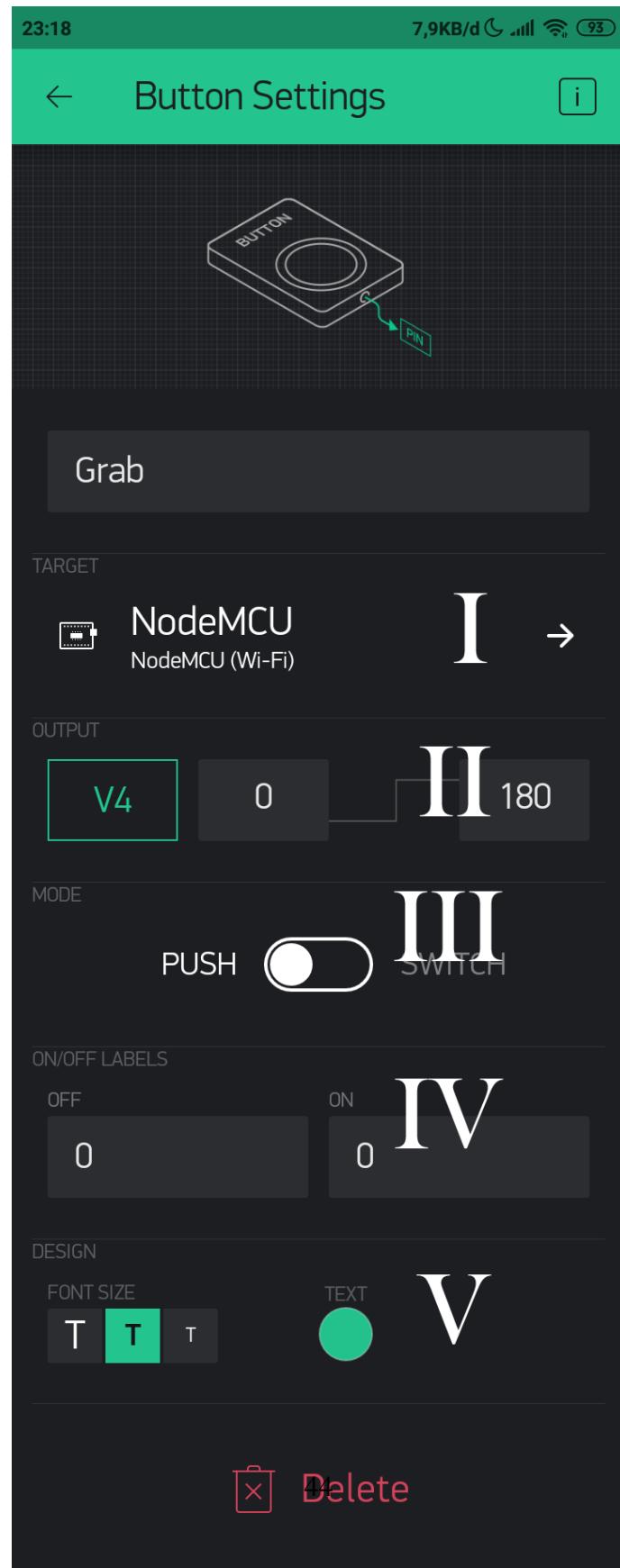


Gambar 32: *Horizontal Slider Settings*

19. Button Settings

Pada tampilan ini merupakan untuk mengatur sebuah tombol. Dalam *project* ini terdapat 2 *button*, *button* pertama di gunakan untuk *Grab*, maka servo akan bergerak dari 180 derajat menuju 0 derajat, sedangkan *button* ke dua di gunakan untuk *drop*, maka servo akan bergerak dari 0 derajat menuju 180 derajat. Berikut penjelasan beberapa bagian yang terdapat pada *Button Settings*:

- I Jenis mikrokontroler yang di gunakan.
- II Jenis pin yang di gunakan, yaitu virtual pin.
- III Pada *button* ini terdapat 2 mode, yaitu mode *switch* dan mode *push*. Jika menggunakan mode *push* ketika *button* tersebut di tekan lalu lepas, maka akan kembali ke posisi awal. Sedangkan jika menggunakan mode *switch*, ketika *button* tersebut di tekan lalu lepas, maka akan tetap di posisi akhir.
- IV Pada bagian ini merupakan untuk pemberian label nama *ON* dan *OFF* pada *button*.
- V Pada bagian ini di gunakan untuk merubah ukuran text menjadi kecil, sedang ataupun besar, dan dapat merubah warna text.

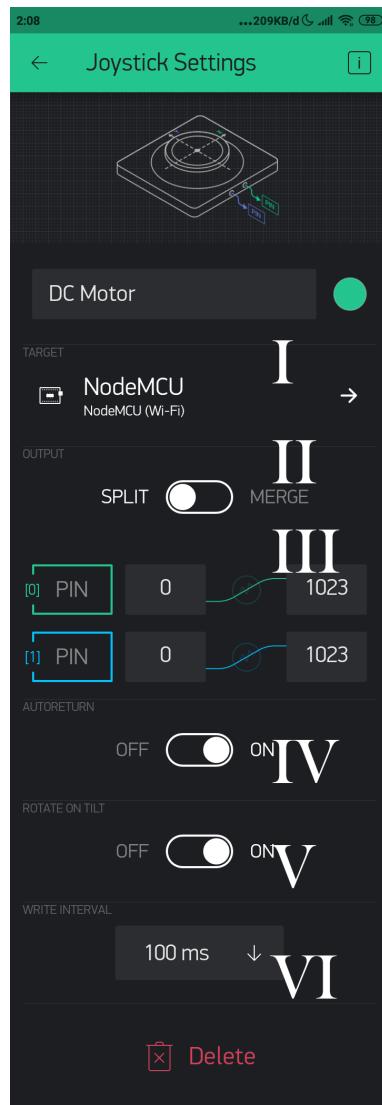


Gambar 33: *Button Settings*

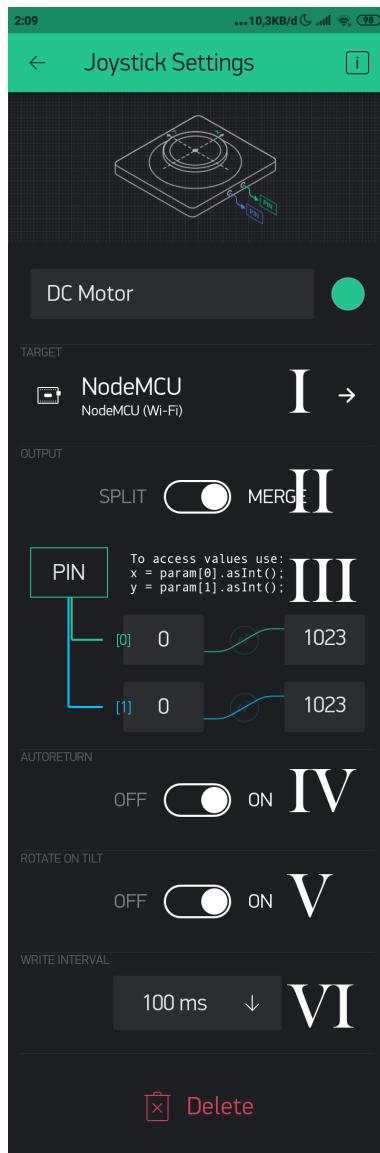
20. Joystick Settings

Pada bagian ini digunakan untuk menggerakan sebuah *DC Motor* dengan menggunakan Joystick. Berikut penjelasan beberapa bagian yang terdapat pada *Joystick settings*:

- I Jenis mikrokontroler yang di gunakan.
- II Pada bagian ini joystick dapat menggunakan mode *SPLIT* atau *MERGE*. Jika menggunakan mode *split* maka terdapat 2 pin yang dapat digunakan. sedangkan jika menggunakan mode *merge* maka hanya terdapat 1 pin yang dapat digunakan. Untuk lebih jelasnya bisa di lihat pada gambar 3.28 dan 3.29.
- III Pada bagian ini ketika *auto return* di aktifkan, maka joystick yang telah di gerakan akan kembali ke posisi awal yaitu di tengah, sedangkan jika *auto return* tidak di aktifkan, maka joystick yang telah di gerakan tidak akan kembali ke posisi awal.
- IV *Rotate on tilt*
- V Pada bagian ini, di gunakan untuk mengirimkan sinyal kecepatan dari blynk menuju servo. Semakin kecil nilai *writing interval* (100 ms) maka akan semakin cepat proses pengiriman sinyal dari blynk ke servo, namun jika semakin besar nilai *writing interval* maka akan semakin lambat dalam proses pengiriman sinyal dari blynk ke servo.



Gambar 34: Joystick Settings dengan mode split

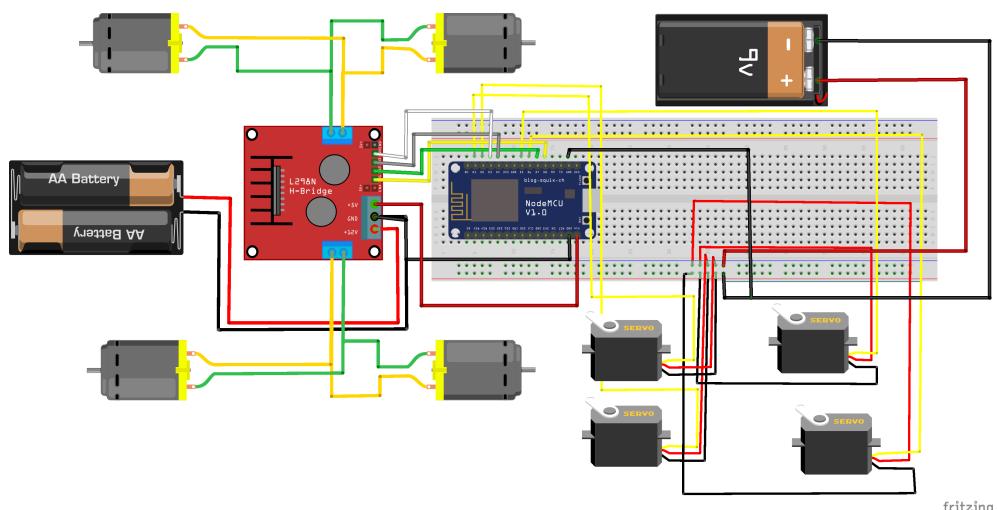


Gambar 35: Joystick Settings dengan mode merge

6 Analisa Rangkaian Secara Detail

Input untuk rangkaian robot lengan adalah blynk pada *smartphone*. NodeMCU dapat diakses pada perangkat smartphone yang akan digunakan sebagai remote control. Ketika analog diarahkan ke atas, maka robot akan melaju ke depan, berlaku juga untuk arah yang lain. Ketika tombol 0 derajat pada remote di tekan, robot akan mengambil barang yang ada di depannya, sedangkan untuk melepaskannya digunakan tombol 180 derajat pada remote.

Kemudian dari modul NodeMCU akan menerima sinyal berupa karakter huruf yang dikirim oleh smartphone menggunakan aplikasinya dan kemudian diteruskan ke L298N. L298N ini akan menerima data dari NodeMCU berupa karakter yang kemudian akan menggerakan Grabber Robot. Setiap komponen pada rangkaian memiliki input karakter huruf yang berbeda yang dihasilkan oleh button pada aplikasi pada smartphone.



Gambar 36: Skematik Rangkaian secara detail

Program untuk joystick

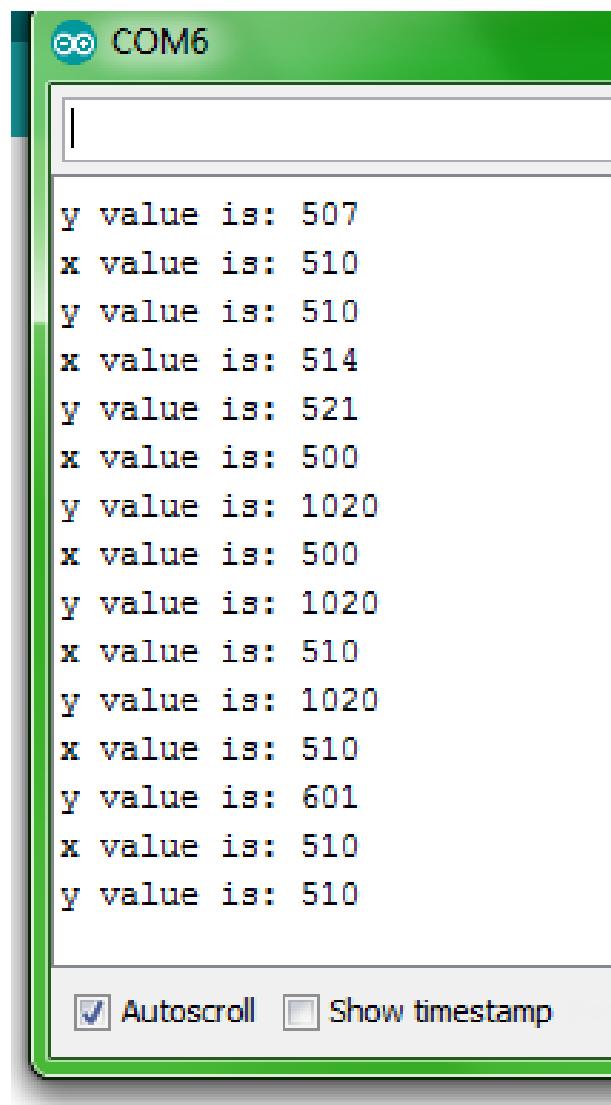
```
BLYNK_WRITE(V6) (1)
{
    int x = param[0].asInt(); (2)
    int y = param[1].asInt();

    Serial.print("x value is: ");
    Serial.println(x);
    Serial.print("y value is: ");
    Serial.println(y);

    moveControl(x, y); (4)
}
```

Pada bagian ini merupakan sebuah program untuk mengendalikan joystick pada motor dc. Berikut beberapa bagian mengenai penjelasan tentang program pada sebuah joystick:

1. Fungsinya sebagai membaca data dari virtual pin.
2. Pada bagian ini merupakan sebuah program untuk joystick yang menggunakan *"Merge"*. Jika menggunakan *"Merge"* maka keluarannya menggunakan 1 pin dan menggunakan *"param"* pada program arduino IDE.
3. Menampilkan nilai joystick yang berupa x dan y, yang dimana nilainya tersebut dapat di lihat pada serial monitor, seperti pada gambar 3.31.
4. Pada bagian ini, x dan y adalah output yang akan menampilkan monitor arduino IDE.



The image shows a screenshot of the Arduino IDE Serial Monitor. The title bar is green with the text "COM6" and a serial port icon. The main window displays a list of text entries. The entries are color-coded: "y value is:" is in blue, "x value is:" is in red, and the numerical values are in black. The text entries are as follows:

```
y value is: 507
x value is: 510
y value is: 510
x value is: 514
y value is: 521
x value is: 500
y value is: 1020
x value is: 500
y value is: 1020
x value is: 510
y value is: 1020
x value is: 510
y value is: 601
x value is: 510
y value is: 510
```

At the bottom of the window, there are two checkboxes: "Autoscroll" (checked) and "Show timestamp" (unchecked).

Gambar 37: Nilai x dan y pada sebuah serial monitor di arduino IDE

Program untuk menjalankan motor dc ketika akan bergerak maju lurus

```
void moveControl( int x , int y) (1)  
{  
    if (y >= maxRange && x >= minRange && x <= maxRange) (2)  
    {  
        digitalWrite (IN_1 , LOW); (3)  
        digitalWrite (IN_2 , HIGH); (4)  
        digitalWrite (IN_3 , HIGH); (5)  
        digitalWrite (IN_4 , LOW); (6)  
    }  
}
```

Pada bagian ini merupakan sebuah program untuk menggerakan motor dc untuk bergerak maju lurus. Berikut penjelasan mengenai beberapa bagian tentang program pada sebuah motor dc :

1. integer bernilai x dan y
2. Output dari program ini akan membuat motor dc bergerak maju lurus.
3. Pin digital ini berjenis PWM dan output dari motor dc 1 akan bernilai *LOW*.
4. Pin digital ini berjenis PWM dan output dari motor dc 2 akan bernilai *HIGH*.
5. Pin digital ini berjenis PWM dan output dari motor dc 3 akan bernilai *HIGH*.
6. Pin digital ini berjenis PWM dan output dari motor dc 4 akan bernilai *LOW*.

Program untuk menjalankan motor dc ketika akan belok kanan

```
else if(x >= maxRange && y >= maxRange) { (1)  
    digitalWrite(IN_1, LOW); (2)  
    digitalWrite(IN_2, HIGH); (3)  
    digitalWrite(IN_3, LOW); (4)  
    digitalWrite(IN_4, HIGH); (5)  
}
```

Pada bagian ini merupakan sebuah program untuk menggerakan motor dc untuk bergerak belok kanan. Berikut penjelasan mengenai beberapa bagian tentang program pada sebuah motor dc :

1. Output dari program ini akan membuat motor dc belok kanan.
2. Pin digital ini berjenis PWM dan output dari motor dc 1 akan bernilai *LOW*.
3. Pin digital ini berjenis PWM dan output dari motor dc 2 akan bernilai *HIGH*.
4. Pin digital ini berjenis PWM dan output dari motor dc 3 akan bernilai *LOW*.
5. Pin digital ini berjenis PWM dan output dari motor dc 4 akan bernilai *HIGH*.

Program untuk menjalankan motor dc ketika akan belok kiri

```
else if(x <= minRange && y >= maxRange){  
    digitalWrite(IN_1, HIGH);  
    digitalWrite(IN_2, LOW);  
  
    digitalWrite(IN_3, HIGH);  
    digitalWrite(IN_4, LOW);  
}
```

Pada bagian ini merupakan sebuah program untuk menggerakan motor dc untuk bergerak belok kiri. Berikut penjelasan mengenai beberapa bagian tentang program pada sebuah motor dc :

1. Output dari program ini akan membuat motor dc belok kiri.
2. Pin digital ini berjenis PWM dan output dari motor dc 1 akan bernilai *HIGH*.
3. Pin digital ini berjenis PWM dan output dari motor dc 2 akan bernilai *LOW*.
4. Pin digital ini berjenis PWM dan output dari motor dc 3 akan bernilai *HIGH*.
5. Pin digital ini berjenis PWM dan output dari motor dc 4 akan bernilai *LOW*.

Program untuk menjalankan motor dc ketika akan berhenti

```
else if(y < maxRange && y > minRange &&  
x < maxRange && x > minRange){           (1)  
    digitalWrite (IN_1 , LOW);                (2)  
    digitalWrite (IN_2 , LOW);                (3)  
    digitalWrite (IN_3 , LOW);                (4)  
    digitalWrite (IN_4 , LOW);                (5)  
}
```

Pada bagian ini merupakan sebuah program untuk menggerakan motor dc untuk berhenti. Berikut penjelasan mengenai beberapa bagian tentang program pada sebuah motor dc :

1. Output dari program ini akan membuat motor dc berhenti.
2. Pin digital ini berjenis PWM dan output dari motor dc 1 akan bernilai *LOW*.
3. Pin digital ini berjenis PWM dan output dari motor dc 2 akan bernilai *LOW*.
4. Pin digital ini berjenis PWM dan output dari motor dc 3 akan bernilai *LOW*.
5. Pin digital ini berjenis PWM dan output dari motor dc 4 akan bernilai *LOW*.

Program untuk menjalankan motor dc ketika akan bergerak mundur

```
else if(y <= minRange && x >= minRange && x <= maxRange){ (1)
    digitalWrite (IN_1, HIGH); (2)
    digitalWrite (IN_2, LOW); (3)
    digitalWrite (IN_3, LOW); (4)
    digitalWrite (IN_4, HIGH); (5)
}
```

Pada bagian ini merupakan sebuah program untuk menggerakan motor dc untuk bergerak mundur. Berikut penjelasan mengenai beberapa bagian tentang program pada sebuah motor dc :

1. Output dari program ini akan membuat motor dc bergerak mundur.
2. Pin digital ini berjenis PWM dan output dari motor dc 1 akan bernilai *HIGH*.
3. Pin digital ini berjenis PWM dan output dari motor dc 2 akan bernilai *LOW*.
4. Pin digital ini berjenis PWM dan output dari motor dc 3 akan bernilai *LOW*.
5. Pin digital ini berjenis PWM dan output dari motor dc 4 akan bernilai *HIGH*.

Program untuk menjalankan motor dc ketika akan bergerak mundur dengan belok kanan

```
else if (y <= minRange && x <= minRange){ (1)  
    digitalWrite (IN_1, LOW); (2)  
    digitalWrite (IN_2, HIGH); (3)  
  
    digitalWrite (IN_3, LOW); (4)  
    digitalWrite (IN_4, HIGH); (5)  
}
```

Pada bagian ini merupakan sebuah program untuk menggerakan motor dc untuk bergerak mundur dengan belok kanan. Berikut penjelasan mengenai beberapa bagian tentang program pada sebuah motor dc :

1. Output dari program ini akan membuat motor dc bergerak mundur dengan belok kanan.
2. Pin digital ini berjenis PWM dan output dari motor dc 1 akan bernilai *LOW*.
3. Pin digital ini berjenis PWM dan output dari motor dc 2 akan bernilai *HIGH*.
4. Pin digital ini berjenis PWM dan output dari motor dc 3 akan bernilai *LOW*.
5. Pin digital ini berjenis PWM dan output dari motor dc 4 akan bernilai *HIGH*.

Program untuk menjalankan motor dc ketika akan bergerak mundur dengan belok kiri

```
else if (y <= minRange && x >= maxRange){ (1)
    digitalWrite (IN_1, HIGH); (2)
    digitalWrite (IN_2, LOW); (3)

    digitalWrite (IN_3, HIGH); (4)
    digitalWrite (IN_4, LOW); (5)
}
```

Pada bagian ini merupakan sebuah program untuk menggerakan motor dc untuk bergerak mundur dengan belok kiri. Berikut penjelasan mengenai beberapa bagian tentang program pada sebuah motor dc :

1. Output dari program ini akan membuat motor dc bergerak mundur dengan belok kiri.
2. Pin digital ini berjenis PWM dan output dari motor dc 1 akan bernilai *HIGH*.
3. Pin digital ini berjenis PWM dan output dari motor dc 2 akan bernilai *LOW*.
4. Pin digital ini berjenis PWM dan output dari motor dc 3 akan bernilai *HIGH*.
5. Pin digital ini berjenis PWM dan output dari motor dc 4 akan bernilai *LOW*.

Tipe data utama untuk penyimpanan bilangan

```
int minRange = 312;  
int maxRange = 712;
```

- **int minRange**

tipe data ini bersifat integer, dengan memiliki *minimal Range* 312 yang digunakan untuk menjalankan motor dc.

- **int maxRange**

tipe data ini bersifat integer, dengan memiliki *maximal Range* 712 yang digunakan untuk menjalankan motor dc.

Library untuk L298N *motor driver*

```
#define IN_1 15          (1)  
#define IN_2 13          (2)  
#define IN_3 2           (3)  
#define IN_4 0           (4)
```

1. Pada l298N ini menggunakan pin IN 1 dan L298N ini terhubung ke Node-MCU dengan menggunakan GPIO 15 (D8)
2. Pada l298N ini menggunakan pin IN 2 dan L298N ini terhubung ke Node-MCU dengan menggunakan GPIO 13 (D7)
3. Pada l298N ini menggunakan pin IN 3 dan L298N ini terhubung ke Node-MCU dengan menggunakan GPIO 2 (D4)
4. Pada l298N ini menggunakan pin IN 4 dan L298N ini terhubung ke Node-MCU dengan menggunakan GPIO 0 (D3)

Program untuk *horizontal slider* yang mengendalikan *base servo*

```
BLYNK_WRITE(V1)          (1)  
{  int a = param.asInt();    (2)  
servo1.write(a);          (3)  
}
```

1. Fungsinya sebagai membaca data dari virtual pin.
2. int a yang artinya nilainya bersifat integer, dan "param" ini di gunakan untuk membaca fungsi dari virtual pin.
3. servo 1 ini adalah *base servo*

Program untuk *vertical slider* yang mengendalikan *elbow servo*

```
BLYNK_WRITE(V2)          (1)  
{  int a = param.asInt();    (2)  
servo2.write(a);          (3)  
}
```

1. Fungsinya sebagai membaca data dari virtual pin.
2. int a yang artinya nilainya bersifat integer, dan "param" ini di gunakan untuk membaca fungsi dari virtual pin.
3. servo 2 ini adalah *elbow servo*

Program untuk *vertical slider* yang mengendalikan *shoulder servo*

```
BLYNK_WRITE(V3)          (1)  
{  int a = param.asInt();    (2)  
servo3.write(a);          (3)  
}
```

1. Fungsinya sebagai membaca data dari virtual pin.
2. int a yang artinya nilainya bersifat integer, dan "param" ini di gunakan untuk membaca fungsi dari virtual pin.
3. servo 3 ini adalah *shoulder servo*

Program untuk button 0 derajat yang mengendalikan *gripper servo*

```
BLYNK_WRITE(V4){          (1)
    int state = param.toInt();    (2)
    if (state == 1) {           (3)
        servo4.write(0);        (4)
    }
}
```

1. Fungsinya sebagai membaca data dari virtual pin.
2. int a yang artinya nilainya bersifat integer, dan "param" ini di gunakan untuk membaca fungsi dari virtual pin.
3. Statement 1 menandakan robot bergerak.
4. servo 4 ini adalah *gripper servo* dan output dari program ini ialah untuk melepaaskan barang yang ada di depannya.

Program untuk button 180 derajat yang mengendalikan *gripper servo*

```
BLYNK_WRITE(V5){          (1)
    int state = param.toInt();    (2)
    if (state == 1) {           (3)
        servo4.write(180);       (4)
    }
}
```

1. Fungsinya sebagai membaca data dari virtual pin.
2. int a yang artinya nilainya bersifat integer, dan "param" ini di gunakan untuk membaca fungsi dari virtual pin.
3. Statement 1 menandakan robot bergerak.
4. servo 4 ini adalah *gripper servo* dan output dari program ini ialah untuk mengambil barang yang ada di depannya.

Program untuk mendeklarasikan servo

```
servo1.attach(5);          (1)  
servo2.attach(4);          (2)  
servo3.attach(14);         (3)  
servo4.attach(12);         (4)  
  
Servo servo1;             (5)  
Servo servo2;             (6)  
Servo servo3;             (7)  
Servo servo4;             (8)
```

1. Servo1.attach ini yang menentukan variabel servo 1 pada GPIO 5 di Node-MCU.
2. Servo2.attach ini yang menentukan variabel servo 2 pada GPIO 4 di Node-MCU.
3. Servo3.attach ini yang menentukan variabel servo 3 pada GPIO 14 di NodeMCU.
4. Servo4.attach ini yang menentukan variabel servo 4 pada GPIO 12 di NodeMCU.
5. Servo.servo1 ini ialah untuk menentukan servo yang digunakan yaitu *base servo*.
6. Servo.servo1 ini ialah untuk menentukan servo yang digunakan yaitu *elbow servo*.
7. Servo.servo1 ini ialah untuk menentukan servo yang digunakan yaitu *shoulder servo*.
8. Servo.servo1 ini ialah untuk menentukan servo yang digunakan yaitu *gripper servo*.

Program untuk menjalankan aplikasi Blynk

```
void setup() (1)  
{  
    Serial.begin(9600); (2)  
    Blynk.begin(auth, ssid, pass); (3)
```

1. void setup

Void Setup ini merupakan semua kode yang ada di void setup akan dibaca sekali oleh Arduino atau mikrokontroller lainnya yang dapat menggunakan software arduino.

2. Serial.begin(9600)

untuk menentukan kecepatan pengiriman dan penerimaan data melalui port serial. Kecepatan yang umum digunakan adalah 9600 bit per detik (9600 bps). Namun, kecepatan hingga 115.200 detik didukung oleh Arduino Uno.

3. Blynk.begin(auth, ssid, pass)

untuk mengaktifkan program IoT pada arduino IDE, agar blynk dapat terhubung ke jaringan WiFi.

Library ini untuk menjalankan *remote control* robot lengan pada aplikasi *blynk*

```
#define BLYNK_PRINT Serial          (1)
#include <ESP8266WiFi.h>           (2)
#include <BlynkSimpleEsp8266.h>      (3)
#include <Servo.h>                 (4)
```

Berikut penjelasan mengenai library yang digunakan untuk menjalankan *remote control* robot lengan pada aplikasi *blynk*:

1. BLYNK _ PRINT Serial

library ini digunakan untuk membaca dan menerima program dari Arduino IDE.

2. ESP8266WiFi.h

library ini digunakan untuk menghubungkan ESP8266 dengan WiFi agar dapat menjalankan *Grabber Robot*.

3. BlynkSimpleEsp8266.h

library ini digunakan untuk menghubungkan ESP8266 dengan Blynk.

4. servo.h

library ini digunakan untuk servo.

Sebagai penghubung antara NodeMCU dengan aplikasi blynk

```
char auth [] = "7d786b75eb914657ba029a78faff50d1";  
char ssid [] = "qwerty";  
char pass [] = "qwerty123";
```

Dalam memprogram IoT menggunakan *remote control blynk* diwajibkan menggunakan auth, ssid, dan pass sebagai penghubung antara program arduino IDE dengan aplikasi blynk.

- **char auth**

Ini merupakan autentikasi yang di dapatkan di bagian *project settings* dan di kirim melalui email. Autentikasi ini berisfat char (*character*), tipe data char adalah tipe data yang mengambil satu byte memori yang menyimpan suatu nilai karakter. Karakter harfiah ditulis dalam kutip tunggal, misalnya 'A' dan untuk multi karakter digunakan tanda kutip ganda, seperti "ABC".

- **char ssid**

Ini merupakan autentikasi yang di dapatkan di bagian *project settings* dan di kirim melalui email. Autentikasi ini berisfat char (*character*), tipe data char adalah tipe data yang mengambil satu byte memori yang menyimpan suatu nilai karakter. Karakter harfiah ditulis dalam kutip tunggal, misalnya 'A' dan untuk multi karakter digunakan tanda kutip ganda, seperti "ABC". **char ssid** ini nama dari WiFi yang digunakan.

- **char pass**

Ini merupakan autentikasi yang di dapatkan di bagian *project settings* dan di kirim melalui email. Autentikasi ini berisfat char (*character*), tipe data char adalah tipe data yang mengambil satu byte memori yang menyimpan suatu nilai karakter. Karakter harfiah ditulis dalam kutip tunggal, misalnya 'A' dan untuk multi karakter digunakan tanda kutip ganda, seperti "ABC". **char pass** ini *password* dari WiFi yang digunakan.

void loop

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

- void loop
 - void loop ini digunakan untuk menjalankan suatu siklus program, yang akan dilakukan terus-menerus hingga mikrokontroler yang digunakan mati/reset.
- Blynk.run
 - Digunakan untuk mengkomunikasikan dengan blynk server.

7 Data Pengamatan

Setelah dilakukan pengujian alat, maka diperoleh hasil berupa data pengamatan sebagai berikut :

Tabel 1: Pengujian Sistem Servo

Jumlah Beban (onz)				
10	20	30	40	50
V	V	X	X	X
V	V	X	X	X
V	V	X	X	X
V	V	X	X	X

Hasil pengujian pada sistem servo diatas menunjukkan bahwa, servo mampu menangkat bebas hanya sampai 20 atau 25 onz saja, lebih dari itu tidak dapat mengangkat benda yang ada di depannya.

Tabel 2: Pengujian sensor jarak koneksi WiFi

Jarak koneksi antara blynk dengan mikrokontroler (M)				
2	5	7	9	10
V	V	V	X	X
V	V	V	X	X
V	V	V	X	X
V	V	V	X	X

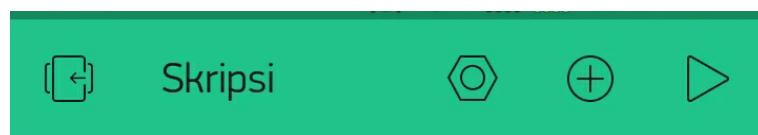
Dari hasil pengujian pada sensor jarak *Wi-Fi* menunjukkan bahwa, robot lengan dapat dikendalikan dengan jarak sejauh 7 Meter. Lebih dari itu tidak dapat terkoneksi, karena menggunakan *Wi-Fi* pada *smartphone*. Berbeda ketika menggunakan *bluetooth*, kalau bluetooth itu hanya bisa sampai 5 meter lebih dari itu akan terjadi putus koneksi.

8 Cara kerja alat

1. Robot dapat bergerak maju dengan menggerakkan analog ke atas pada remote control.
2. Robot dapat bergerak mundur dengan menggerakkan analog ke bawah pada remote control.
3. Robot dapat bergerak ke kiri dengan menggerakkan analog ke kiri pada remote control.
4. Robot dapat bergerak ke kanan dengan menggerakkan analog ke kanan pada remote control.
5. Robot dapat mengambil benda dengan menekan tombol 0 derajat pada remote control.
6. Robot dapat melepaskan benda dengan menekan tombol 180 derajat pada remote control.

9 Langkah-langkah Persiapan Uji Coba

1. Hubungkan baterai 3,7V sebagai aktivator servo dan baterai 9V sebagai aktivator L298N.
2. Lalu aktifkan tethering pada smartphone, kemudian jalankan aplikasi Blynk dan tekan tombol segitiga di bagian pojok kanan atas.
3. Modul NodeMCU akan menyala. Setelah modul NodeMCU menyala, kendalikan robot melalui smartphone, dengan cara menggerakkan analog pada aplikasi robot lengan.



Gambar 38: Kondisi *blynk* belum di jalankan

10 Hasil Uji Coba

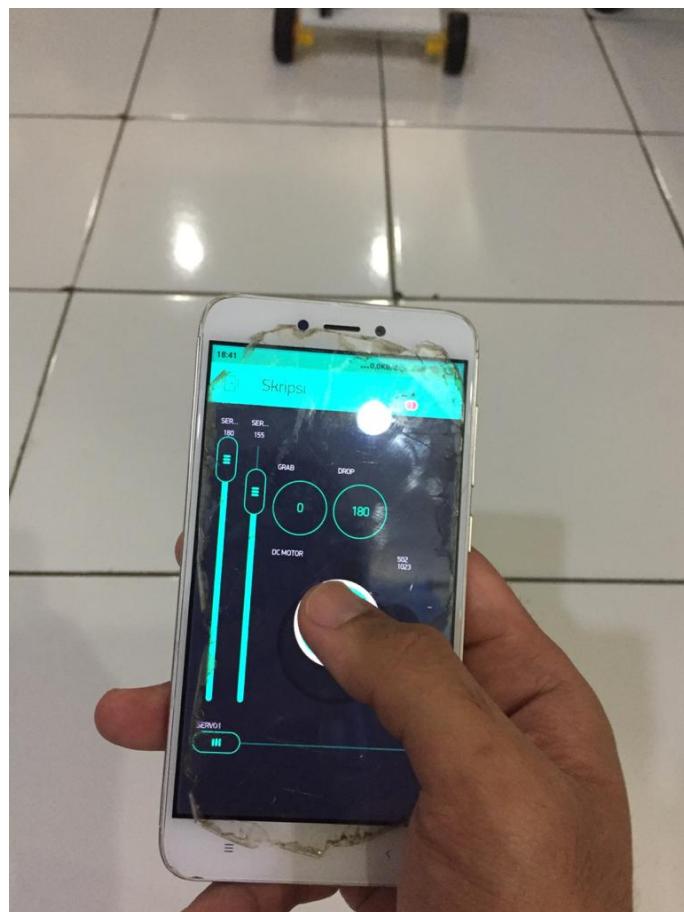
Pengujian yang dilakukan pada prototype sistem kendali lengan robot ini bertujuan agar sistem yang dibuat dapat bekerja sebagaimana mestinya. Dalam proses pengujian ini dibagi menjadi beberapa bagian, yaitu pengujian tombol navigasi, pengujian kesesuaian gerak dengan tombol navigasi, dan pengujian keseluruhan prototype lengan robot.

Berikut adalah pengujian menyalakan NodeMCU dan L298N menggunakan baterai. Proses yang dilakukan adalah menghubungkan dengan baterai, maka motor dc akan bergerak sebentar, lampu led pada NodeMCU dan L298N akan menyala yang menandakan bahwa robot lengan siap digerakkan.



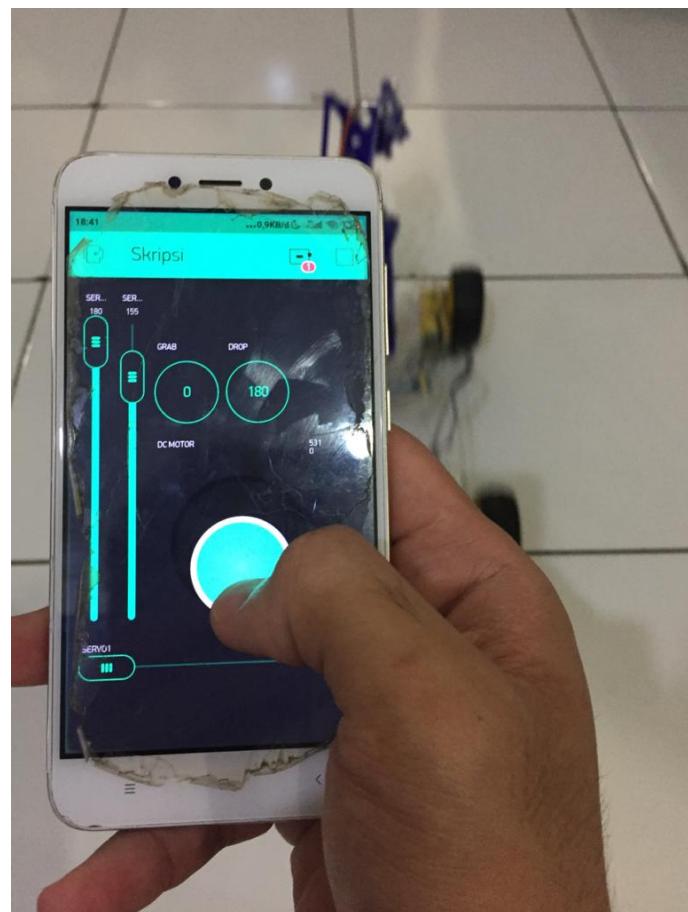
Gambar 39: Robot telah terkoneksi dengan blynk server

Berikut adalah pengujian tombol navigasi pada *joystick*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot lengan ingin bergerak maju lurus, maka *joystick* pada *remote control blynk* dapat di gerakkan ke atas.



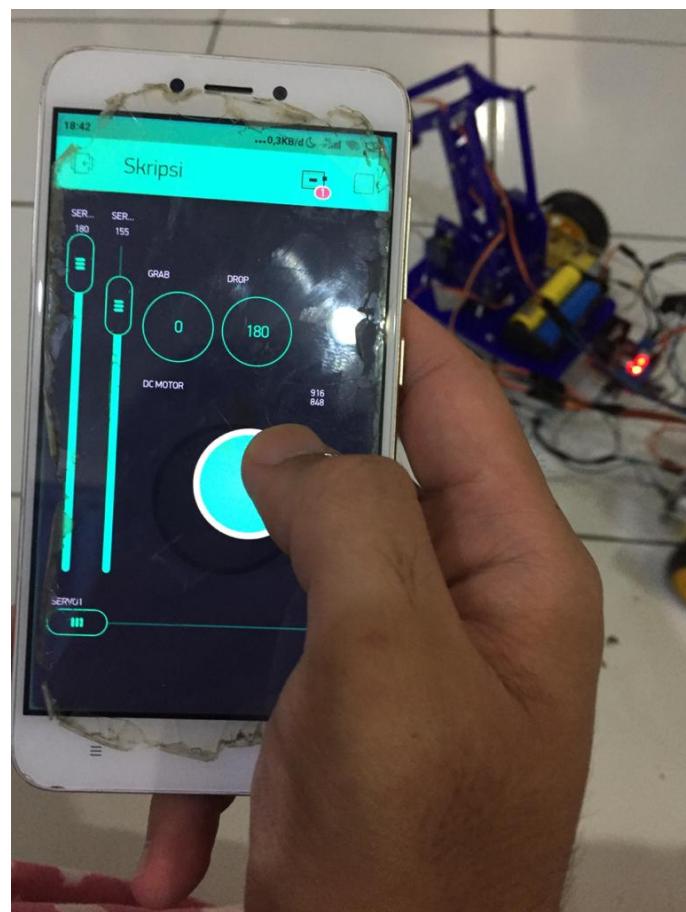
Gambar 40: Robot bergerak mundur

Berikut adalah pengujian tombol navigasi pada *joystick*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot ingin bergerak mundur lurus, maka *joystick* pada *remote control blynk* dapat di gerakkan ke bawah.



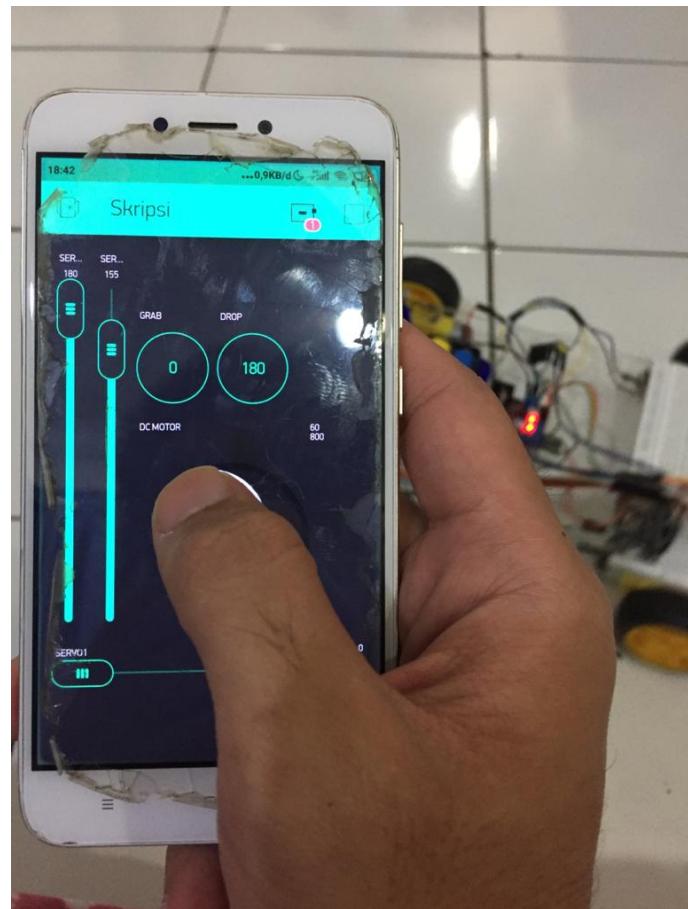
Gambar 41: Robot bergerak mundur

Berikut adalah pengujian tombol navigasi pada *joystick*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot ingin belok kanan, maka *joystick* pada *remote control blynk* dapat di gerakkan ke kanan.



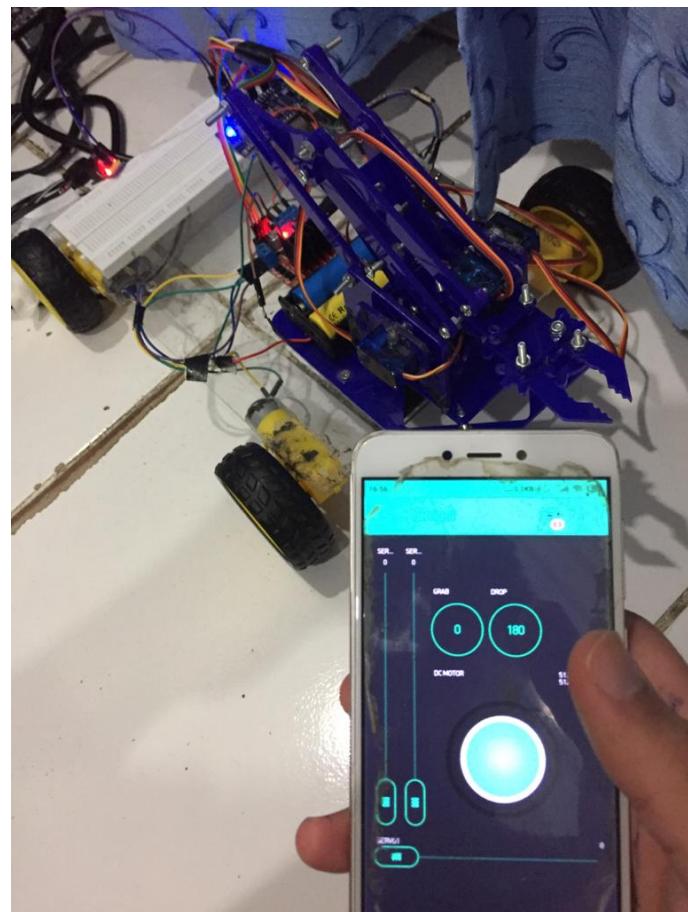
Gambar 42: Robot belok kanan

Berikut adalah pengujian tombol navigasi pada *joystick*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot ingin belok kiri, maka *joystick* pada *remote control blynk* dapat di gerakkan ke kiri.



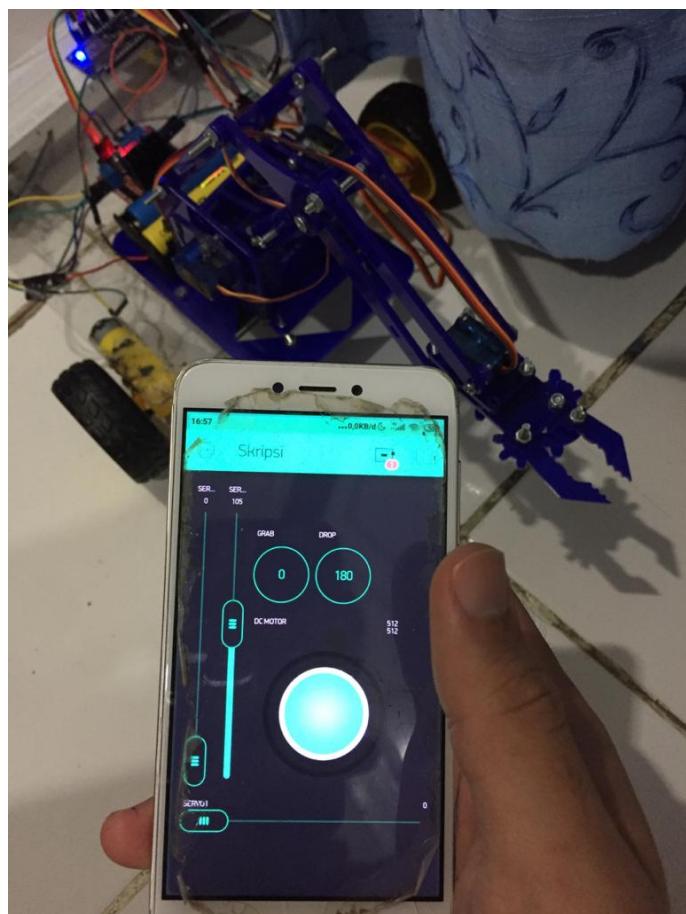
Gambar 43: Robot belok kiri

Ketika robot dihubungkan dengan baterai, lampu led pada NodeMCU dan L298N akan menyala yang menandakan bahwa robot lengan siap digerakkan.



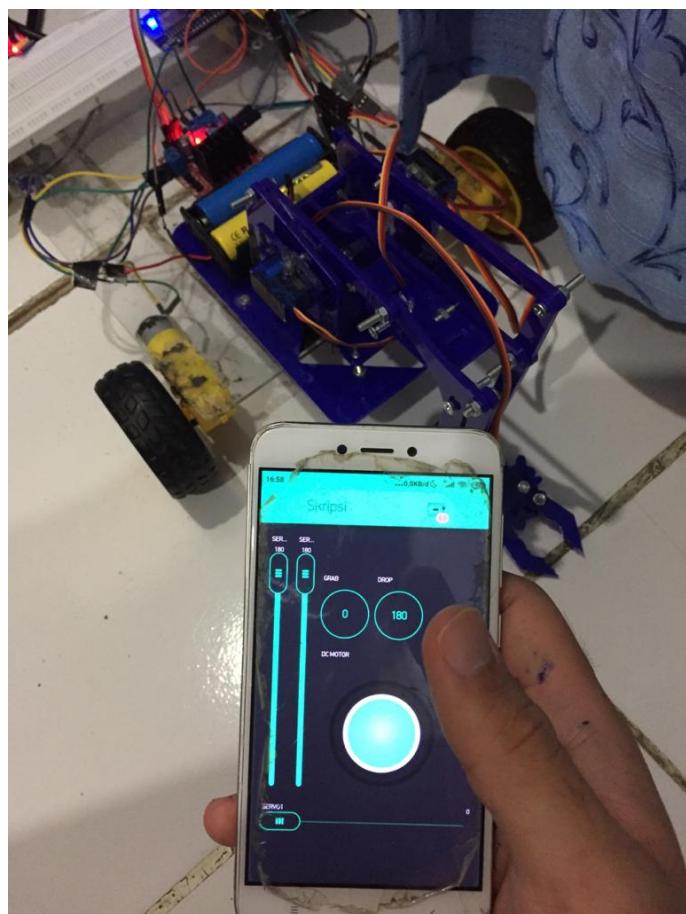
Gambar 44: Servo dalam posisi belum digerakan

Berikut adalah pengujian pengujian kesesuaian gerak servo pada *slider*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot lengan ingin turun atau *shoulder servo* berputar 105 derajat, maka *slider 3* pada *remote control blynk* dapat di geser ke atas atau di geser ke arah 105 derajat.



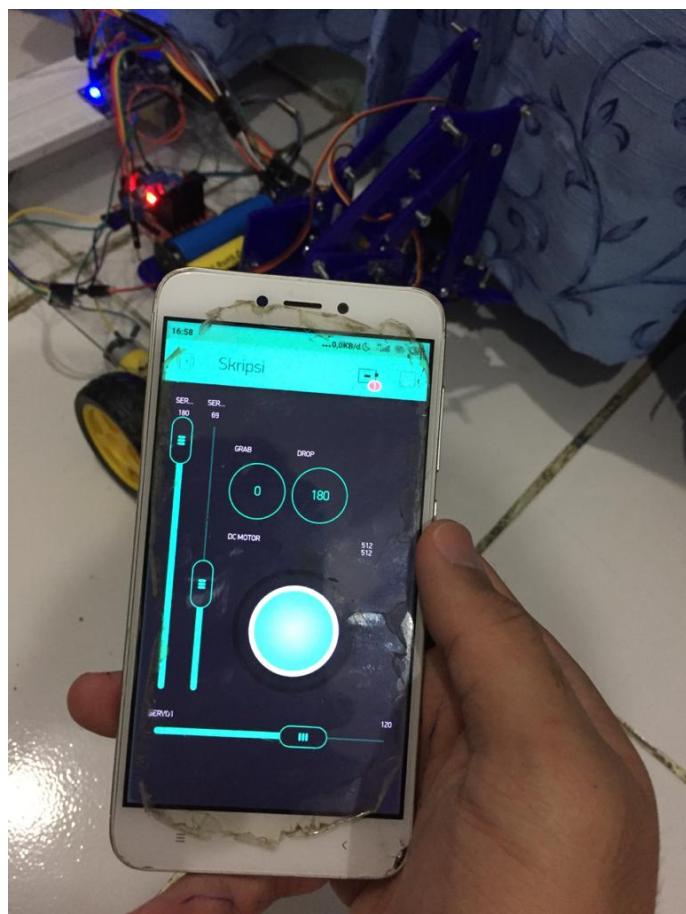
Gambar 45: *Shoulder servo* berputar 105 derajat

Berikut adalah pengujian pengujian kesesuaian gerak servo pada *slider*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot lengan ingin bergerak ke belakang atau elbow servo berputar 180 derajat, maka *slider 2* pada *remote control blynk* dapat di geser ke atas atau di geser ke arah 180 derajat.



Gambar 46: *Elbow servo* berputar 180 derajat

Berikut adalah pengujian pengujian kesesuaian gerak servo pada *slider*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot lengan ingin berputar ke samping atau base servo berputar 120 derajat, maka *slider 1* pada *remote control blynk* dapat di geser ke samping atau di geser ke arah 120 derajat.



Gambar 47: Base servo berputar 180 derajat

Berikut adalah pengujian pengujian kesesuaian gerak servo pada *button*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot lengan ingin mengambil cabai yang ada di depannya atau *gripper servo* berputar 0 derajat, maka *button 0* pada *remote control blynk* dapat di tekan.



Gambar 48: *Gripper servo* berputar 0 derajat

Berikut adalah pengujian pengujian kesesuaian gerak servo pada *button*, untuk memberikan perintah menggerakan robot lengan. Proses yang dilakukan adalah ketika robot lengan ingin melepaskan cabai yang ada di depannya atau *gripper servo* berputar 180 derajat, maka *button 180* pada *remote control blynk* dapat di tekan.



Gambar 49: *Gripper servo* berputar 180 derajat

11 Kesimpulan

Berdasarkan perancangan dan pengujian yang telah dilakukan, penulis dapat menyimpulkan, diantaranya:

1. Grabber Robot yang dikendalikan melalui Blynk untuk membantu memindahkan benda yang ringan berbasis Arduino dan aplikasi android berhasil dibuat.
2. Kondisi output motor Servo sudah sesuai dengan output yang dirancang oleh penulis.

12 Saran

Pada pembuatan robot ini, dapat diambil beberapa saran untuk dikembangkan lebih lanjut, diantaranya :

- Mengganti motor DC dengan motor yang memiliki torsi lebih besar, serta menambah catu daya dengan tegangannya yang lebih besar menyesuaikan bobot beban yang lebih besar misalnya 1kg atau lebih.

Daftar Pustaka

- [1] Spong, Mark W. Hutchinson, Seth. Vidyasagar, M. 2004. *Robot Dynamics and Control Second Edition*.
- [2] Spong, Mark W. Hutchinson, Seth. Vidyasagar, M. 2004. *ESP8266 Node-MCU WiFi Devkit*.
- [3] Hanif Azzam, Ramli Kalamullah, 2015. PERANCANGAN DAN ANALISIS SISTEM KENDALI LENGAN ROBOT BERBASIS MOTION CONTROL. Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia.
- [4] Saifuddin Arief, Sumardi, and Darjat, 2017. PERANCANGAN SISTEM KENDALI PERGERAKAN ARM MANIPULATOR BERBASIS SENSOR INERTIAL MEASUREMENT UNIT (IMU) DAN SENSOR FLEX. Departemen Teknik Elektro, Universitas Diponegoro: TRANSIENT, VOL. 6, NO. 3, SEPTEMBER 2017, ISSN: 2302-9927, 425.
- [5] Setiadi David, Muhaemin Abdul Nurdin Muhamad, 2018. PENERAPAN INTERNET OF THINGS (IoT) PADA SISTEM MONITORING IRIGASI (SMART IRIGASI) . STMIK Sumedang, Universitas Sangga Buana YPKP: Jurnal Infotronik. p-ISSN : 2548-1932 e-ISSN : 2549-7758.
- [6] Muhammad, Syahwil, 2013. Panduan Mudah Simulasi & Praktek Mikrokontroler Arduino. Yogyakarta: Andi Yogyakarta. ISBN: 978-979-29-4102-9.

- [7] Afridani, Wildian, 2015. RANCANG BANGUN SISTEM KENDALI ROBOT TANGAN MENGGUNAKAN BLUETOOTH BERBASIS MIKRO-KONTROLER ATmega8535. Universitas Andalas. ISSN 2302-8491.
- [8] Afif, Thowil, Muhammad ; Ayu, Putri, Pratiwi, Ilham, 2015. ANALISIS PERBANDINGAN BATERAI *LITHIUM-ION*, *LITHIUM-POLYMER*, *LEAD ACID* DAN *NICKEL-METAL HYDRIDE* PADA PENGGUNAAN MOBIL LISTRIK - REVIEW . Malang: Jurnal Rekayasa Mesin. ISSN 2477-6041.
- [9] Nurgraha Rifki, Susanto Erwin, Wibawa Dwi Prasetya Ig., 2016. DESAIN DAN IMPLEMENTASI SISTEM KENDALI LENGAN ROBOT MENGGUNAKAN *LEAP MOTION*. Universitas Telkom: e-Proceeding of Engineering. ISSN : 2355-9365.
- [10] Nugraha Wiria Deny, PERANCANGAN SISTEM KONTROL ROBOT LENGAN YANG DIHUBUNGKAN DENGAN KOMPUTER. Majalah Ilmiah Mektek, SEPTEMBER 2010, hlm 181.
- [11] Kho Dickson, 2020. Pengertian USB (*Universal Serial Bus*) dan Jenis-jenis Konektor USB. <https://teknikelektronika.com/pengertian-usb-universal-serial-bus-jenis-jenis-konektor-usb/>. Diakses pada 28 Juni 2020.
- [12] N Sora, 2017. Pengertian WiFi Dan Fungsinya Maupun Cara Kerjanya. <http://www.pengertianku.net/2017/08/pengertian-wifi-dan-fungsinya-maupun-cara-kerjanya.html> . Diakses pada 28 Juni 2020.
- [13] Harjono, 2017. Fungsi Address Resolution Protocol dalam Ethernet (*Address Resolution Protocol Functions in The Ethernet*). Teknik Informatika, Universitas Muhammadiyah. JUITA p-ISSN: 2086-9398; e-ISSN: 2579-8901.
- [14] Anonim, 2018. BERKENALAN DENGAN GSM, PENGERTIAN, SEJARAH, SERTA FUNGSINYA. https://www.baktikominfo.id/en/informasi/pengetahuan/berkenalan_dengan_gsm_pengertian_sejarah_serta_fungsinya-671. Diakses pada 28 Juni 2020.
- [15] Indrayana Sukma Arief, Primananda Rakhmadhany, Amron Kasyful, 2018. Rancang Bangun Sistem Komunikasi Bluetooth Low Energy(BLE) Pada Sistem Pengamatan Tekanan Darah. Teknik Informatika, Universitas Brawijaya. e-ISSN: 2548-964X.

- [16] CodeChamp. Interface L298N Using NodeMCU. <https://www.instructables.com/id/Interface-L298N-Using-NodeMCU/>. Diakses pada 28 Juni 2020.
- [17] Saputro TRI TEDY, 2017. Mengenal NodeMCU: Pertemuan Pertama. Embeddednesia. <https://embeddednesia.com/v1/tutorial-nodemcu-pertemuan-pertama/>. Diakses pada 28 Juni 2020.
- [18] Didi Martinus, Marindani Derdian Elang, Elbani Ade. Rancang Bangun Pengendalian Robot Lengan 4 DOF dengan GUI (Graphical User Interface) Berbasis Arduino Uno. Fakultas Teknik, Universitas Tanjungpura.
- [19] Faudin, Agus.2017. Apa itu Module NodeMCU ESP8266?. <https://www.nyebarilmu.com/apa-itu-module-nodemcu-esp8266/>. Diakses 17 Oktober 2019.
- [20] Anonim.2014. Apa itu Module NodeMCU ESP8266?. <http://zonaelektront.net/motor-dc/>. Diakses 7 April 2018.
- [21] Silego, Martinez Antonio, 2017. *Creating a Smart Home with Blynk*. <https://www.allaboutcircuits.com/industry-articles/creating-a-smart-home-with-blynk/>. Diakses pada 28 Juni 2020.
- [22] Caysar Dina, 2018. PENGATURAN PERGERAKAN ROBOT LENGAN SMART ARM ROBOTIC AX-12A MELALUI PENDEKATAN GEOMETRY BASED KINEMATIC MENGGUNAKAN ARDUINO. Teknik Elektro, Universitas Brawijaya.
- [23] P Ary Kade Ngurah Gusti I, Ekayana Gde Agung Anak, 2017. RANCANG BANGUN *PROTOTYPE SISTEM KENDALI LENGAN ROBOT MENGGUNAKAN INTERFACE WIRELESS 2.4Ghz*. Teknik Elektro, Universitas Pendidikan Ganesha Singaraja. P-ISSN : 2303-3142 E-ISSN : 2548-8570
- [24] Nawale Balasaheb, 2020. *3 Steps to build your IoT Project with Blynk IoT app [updated]*. <https://iotdunia.com/blynk-iot/>. Diakses pada 28 Juni 2020.
- [25] Pavel, 2020. *What is Virtual Pins*. <http://help.blynk.cc/en/articles/512061-what-is-virtual-pins>. Diakses pada 28 Juni 2020.
- [26] Koyanagi Fernando, 2020. *NodeMCU ESP8266: Details and Pinout*. <https://www.instructables.com/id/NodeMCU-ESP8266-Details-and-Pinout/>. Diakses pada 28 Juni 2020.

- [27] Pavel, 2020. *How to control anything with Blynk app.* <http://help.blynk.cc/en/articles/512059-how-to-control-anything-with-blynk-app>. Diakses pada 28 Juni 2020.