

***TRAINER KIT CONTROL SYSTEM AND DATA ACQUISITION***

**PROYEK AKHIR**



Diusul oleh :

Gazali Ma'rif

NIM : 0031811

Winda Riani

NIM : 0031829

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2021**

**LEMBAR PENGESAHAN**

**JUDUL PROYEK AKHIR**

***TRAINER KIT CONTROL SYSTEM AND DATA ACQUISITION***

Oleh

Gazali Ma'rif

NIRM : 0031811

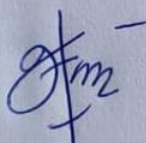
Winda Riani

NIRM : 0031829

Makalah ini telah disetujui dan disahkan oleh pembimbing sebagai salah satu syarat kelulusan Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

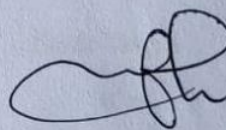
Menyetujui

Pembimbing 1



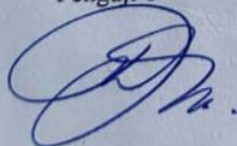
I Made Andik Setlawan, M.Eng, Ph.D

Pembimbing 2



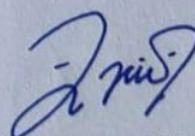
Dr. Parulian Silalahi, M.Pd

Penguji 1



Indra Dwisaputra, M.T

Penguji 2



Zanu Saputra, M.Tr.T

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Winda Riani NIM : 0031829

Nama Mahasiswa 2 : Gazali Ma'rif NIM : 0031818

Dengan Judul : *TRAINER KIT CONTROL SYSTEM AND DATA  
ACQUISITION*

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat 9 Agustus 2021

Nama Mahasiswa

Tanda Tangan

1. Gazali Ma'rif



2. Winda Riani



## **ABSTRAK**

*Untuk mendukung serta menunjang kegiatan belajar mengajar yang ada di Politeknik Manufaktur Negeri Bangka Belitung, maka setiap pengajar harus dapat mempermudah dalam memberikan simulasi pembelajaran secara verbal kepada mahasiswa. Salah satu pembelajaran yang dapat diambil adalah mengenai control system dan akuisisi data. Mahasiswa dapat dengan mudah untuk melakukan pengontrolan dan akuisisi data melalui smartphone yang mereka miliki masing-masing. Trainer kit ini dibuat sebagai pengembangan dari pemanfaatan sistem SCADA berbasis mobile. Terdapat beberapa komponen umum seperti LED, sensor tegangan, sensor INA219, sensor DHT22, serta driver motor & motor yang dapat dilakukan pengontrolan serta akuisisi data melalui trainer kit yang dapat dihubungkan ke smarthphone melalui jaringan internet yang biasa disebut dengan sistem IoT. Komponen LED menampilkan pengontrolan nyala dan mati LED, sensor tegangan menampilkan nilai pengukuran tegangan, sensor INA219 menampilkan nilai arus dan tegangan, sensor DHT22 menampilkan nilai suhu dan kelembaban, driver motor & motor menampilkan pengontrolan motor. Keluaran yang dihasilkan ditampilkan pada platform blynk.*

***Kata kunci : Trainer Kit, SCADA, Iot, Blynk***

## **ABSTRACT**

*To support and development teaching and learning activities at Bangka Belitung State Manufacturing Polytechnic, every teacher must be able to make it easier to provide verbal learning simulations to collage students. One of the lesson that can be taken is about the control system and data acquisition. Every collage students can easily control and acquire data through their respective smartphones. This trainer kit was created as a development of the use of a mobile-based SCADA system. There are several general components such as LEDs, voltage sensors, INA219 sensors, DHT22 sensors, and motors with drivers motor that can be controlles and acquired data through a trainer kit that can be connected to a smartphone via the internet network which is commonly referred to as an IoT system. The LED component displays the LED on and off control, the voltage sensor displays the voltage measurement value, the INA219 sensor displays the current and voltage values, the DHT22 sensor displays the temperature and humidity values, the motors eith drivers motor displays the motor control. The resulting output is displayed on the blynk platform.*

**Keywords :** *Trainer Kit, SCADA, IoT,Blynk*

## KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT yang telah melimpahkan banyak sekali rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan makalah proyek akhir dengan judul “ *Trainer Kit Control System and Data Acquisitoin*”, sebagai salah satu syarat untuk menyelesaikan Program D-III jurusan Elektronika.

Pada kesempatan kali ini, izinkan penulis untuk menyampaikan ucapan banyak terimakasih yang setulus-tulusnya kepada :

1. Allah SWT karena telah mempermudah jalan penulis dalam penyelesaian makalah proyek akhir ini.
2. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku pembimbing I yang telah bersedia meluangkan waktunya untuk memberikan arahan serta saran untuk penyelesaian makalah proyek akhir ini.
3. Bapak Dr.Parulian Silalahi, M.pd selaku dosen pembimbing II yang telah membantu memberikan saran-saran dalam pembuatan makalah proyek akhir ini.
4. Bapak Ocsirendi, M.T selaku wali dosen kelas 3 Elektronika A yang telah membimbing serta memberikan saran.
5. Seluruh staff pengajar jurusan Elektronika dan Informatika yang telah memberikan pembelajaran serta ilmu pengetahuan yang tidak ternilai harganya selama penulis menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung.
6. Keluarga penulis yang telah menjadi *support system*, memberikan kasih sayang, serta doa untuk penulis dalam menyelesaikan makalah proyek akhir ini.
7. Pihak-pihak yang ikut andil dalam membantu penulis dalam penyelesaian makalah proyek akhir ini.

Penulis sangat menyadari bahwa banyak sekali kekurangan pada makalah ini, dan tidak mungkin akan terselesaikan tanpa adanya dukungan, bimbingan, serta nasehat dari berbagai pihak selama proses pembuatan makalah maupun alat proyek akhir ini. Maka dari itu, kritik dan saran sangat diharapkan agar kedepannya penulis dapat lebih mengembangkan penulisan selanjutnya.

Akhir kata, penulis ucapkan terimakasih. Semoga makalah ini dapat bermanfaat untuk pembaca serta penulis.

Sungailiat, 9 Agustus 2021

Penulis

## DAFTAR ISI

LEMBAR PERSETUJUAN PROPOSAL .....	<b>Error! Bookmark not defined.</b>
PERNYATAAN BUKAN PLAGIAT .....	ii
ABSTRAK .....	iii
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR .....	xi
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Batasan Masalah.....	2
1.3 Perumusan Masalah.....	2
1.4 Tujuan.....	2
BAB II DASAR TEORI .....	3
2.1 SCADA .....	3
2.2 Sistem Kontrol <i>Open Loop &amp; Close Loop</i> .....	3
2.3 <i>Trainer kit</i> .....	4
2.4 IOT .....	4
2.5 Arduino IDE .....	5
2.6 ESP32 .....	6
BAB III METODE PELAKSANAAN .....	7
3.1 Pengelompokkan Jenis Praktikum.....	7
3.2 Tahapan Pelaksanaan .....	8
3.3 Penggunaan Platform Blynk.....	8
BAB IV PEMBAHASAN.....	10
4.1 Uji Coba Tiap-Tiap Komponen.....	10
4.1.1 Wemos DI-R32 .....	10



4.1.2	LED .....	11
4.1.3	<i>Switch</i> .....	17
4.1.4	Sensor DHT22.....	20
4.1.5	Sensor INA219.....	25
4.1.6	Sensor Tegangan .....	31
4.1.7	<i>Driver</i> Motor dan Motor .....	37
4.2	Desain Trainer Kit .....	42
4.3	Layout PCB Secara Keseluruhan Komponen .....	43
4.4	Perakitan <i>Casing</i> .....	45
4.5	Kuisisioner .....	45
BAB V KESIMPULAN DAN SARAN.....		48
5.1	Kesimpulan.....	48
5.2	Saran .....	48
DAFTAR PUSTAKA .....		50

## DAFTAR TABEL

Tabel 4.1 Hasil Pengukuran Sensor DHT22 .....	23
Tabel 4.2 Hasil Nilai Tegangan dan Arus.....	27
Tabel 4.3 Tabel hasil pengujian sensor tegangan.....	34
Tabel 4.4 Tabel Kecepatan Motor.....	41
Tabel 4.5 Persentase Jawaban Kuisiner No.1 .....	46
Tabel 4.6 Persentase Jawaban Kuisiner No.2 .....	46
Tabel 4.7 Persentase Jawaban Kuisiner No.3 .....	46
Tabel 4.8 Persentase Jawaban Kuisiner No.4 .....	46
Tabel 4.9 Persentase Jawaban Kuisiner No.5 .....	47

## DAFTAR GAMBAR

Gambar 2.1 <i>Open Loop Control</i> [4].....	4
Gambar 2.2 <i>Close Loop Control</i> [4] .....	4
Gambar 2.3 Arduino IDE.....	5
Gambar 2.4 Modul Wemos D1-R32 [6] .....	6
Gambar 3.1 Blok Diagram <i>Trainer Kit</i> .....	7
Gambar 3.2 Flowchart Tahapan Pelaksanaan .....	8
Gambar 4.1 Pemasangan Modul Wemos DI-R32.....	10
Gambar 4.2 Gambar Rangkaian Pengontrolan LED.....	11
Gambar 4.3 Pemasangan LED Pada Papan PCB.....	12
Gambar 4.4 Tampilan Desain LED Blynk.....	15
Gambar 4.5 Lampu 1 Menyala .....	15
Gambar 4.6 Lampu 2 Menyala .....	16
Gambar 4.7 Lampu 3 Menyala .....	16
Gambar 4.8 Lampu 4 Menyala .....	16
Gambar 4.9 Rangkaian Switch.....	17
Gambar 4.10 Pemasangan <i>Switch</i> .....	18
Gambar 4.11 Tampilan Blynk Pengontrolan <i>Switch</i> .....	19
Gambar 4.12 Rangkaian Sensor Suhu.....	20
Gambar 4.13 Pemasangan Sensor DHT22 Pada Papan PCB.....	20
Gambar 4.14 Hasil Suhu & Kelembaban Pada Blynk .....	22
Gambar 4.15 Pengukuran Suhu & Kelembaban Dengan Alat Ukur.....	23
Gambar 4.16 Grafik Kestabilan Sensor DHT22 .....	24
Gambar 4.17 Rangkaian Sensor INA219.....	25
Gambar 4.18 Pemasangan Sensor Arus INA219 Pada Papan PCB .....	26
Gambar 4.19 Tampilan Output Sensor Arus INA219.....	27
Gambar 4.20 Grafik Arus Sensor INA219.....	29
Gambar 4.21 Grafik Tegangan Sensor INA219.....	30
Gambar 4.22 Rangkaian Sensor Tegangan .....	31

Gambar 4.23 Pemasangan Sensor Tegangan Pada Papan PCB .....	32
Gambar 4.24 Hasil Sensor Tegangan.....	34
Gambar 4.25 Grafik Hasil Sensor Tegangan .....	35
Gambar 4.26 Rangkaian <i>Driver</i> Motor & Motor.....	37
Gambar 4.27 Pemasangan <i>Driver</i> Motor & Motor .....	38
Gambar 4.28 Tampilan Pada Blynk.....	41
Gambar 4.29 Grafik Tegangan Motor.....	42
Gambar 4.30 Desain Trainer Kit.....	43
Gambar 4.31 Desain Layout PCB 2D.....	44
Gambar 4.32 Hasil Pemasangan Secara Keseluruhan.....	44
Gambar 4.33 Desain <i>Casing</i> .....	45

## DAFTAR LAMPIRAN

- Lampiran 1 : Daftar Riwayat Hidup.
- Lampiran 2 : Program Tiap-Tiap Komponen.
- Lampiran 3 : Penggunaan Platform Blynk Pada Tiap-Tiap Praktikum.
- Lampiran 4 : Kuisisioner *Trainer Kit Control System and Data Acquisition*.
- Lampiran 5 : Modul Praktikum

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Fasilitas penunjang untuk pembelajaran tentu sangat dibutuhkan. Terlebih untuk menghadapi kemajuan teknologi yang semakin pesat. Pengajar diminta untuk bisa memanfaatkan dan mengelola hasil dari perkembangan teknologi sebagai alat penunjang dalam belajar-mengajar. Salah satu kemajuan teknologi yang dapat digunakan yaitu SCADA (*Supervisory Control And Data Acquisition*) yang mana merupakan konsep pengendalian industri bertema komputer yang digunakan untuk melakukan kontrol suatu proses contohnya industri, infrastruktur, serta fasilitas [1]. Teknologi lain yang sedang marak digunakan dan sangat membantu penggunanya yaitu IOT (*Internet of Things*) yang merupakan kegiatan mengirim data dengan jaringan tanpa perlu interaksi antar manusia atau dari manusia ke komputer [2].

Adapun melihat sistem pembelajaran di Politeknik Manufaktur Negeri Bangka Belitung terutama pada pembelajaran praktikum masih belum terbilang efektif. Maka dari itu penggunaan alat peraga yang sederhana namun efisien memudahkan seorang pengajar dalam melakukan pengoperasian alat tersebut hingga tujuan belajar mengajar dapat dicapai.

Salah satu cara agar tujuan pembelajaran dapat tercapai, maka dibuatlah suatu alat yang dapat membantu pembelajaran tentang sistem kontrol dan data akuisisi, yang mana pembelajaran tersebut berguna untuk pengontrolan dan akuisisi data yang biasa digunakan pada industri untuk mempermudah suatu pekerjaan dengan menampilkan grafis pada software.

## 1.2 Batasan Masalah

Batasan masalah dari pembuatan proyek akhir ini adalah :

- *Trainer kit* ini mencakup pembelajaran sistem kontrol dan akuisisi data khususnya pada komponen LED, *switch*, sensor DHT22, sensor INA219, sensor tegangan, serta *driver* motor & motor.

## 1.3 Perumusan Masalah

- Bagaimana cara mengembangkan *trainer kit control system and data acquisition* agar dapat digunakan oleh mahasiswa Polman Babel ?
- Bagaimana cara membuat *trainer kit control system and data acquisition* yang dapat mengeluarkan *output* secara efektif berupa nilai serta grafik ?

## 1.4 Tujuan

- Membuat alat peraga yang mencakup pembelajaran *control system and data acquisition*.
- Dapat menghasilkan *output* dari pengontrolan berupa nilai serta grafik pada aplikasi sehingga dapat dilakukan pengelompokan data.

## **BAB II**

### **DASAR TEORI**

Beberapa dasar teori dibawah ini diperoleh dari berbagai referensi berupa internet dan buku-buku. Berikut teori yang digunakan untuk menyelesaikan proyek akhir ini.

#### **2.1 SCADA**

SCADA merupakan pengertian dari monitoring serta pengelompokkan data yang artinya mengawasi, mengendalikan dan melakukan pengumpulan data, mengumpulkan informasi, lalu mentransfer kembali ke pusat, serta menampilkan data pada display. Biasanya, SCADA digunakan untuk kontrol suatu proses seperti kegiatan pada industri mencakup manufaktur serta produksi. SCADA juga digunakan pada proses pembangunan [3].

Salah satu contoh pengaplikasian SCADA biasanya terdapat pada proses industri sebagai contoh industri migas. Aplikasi SCADA dapat digunakan sebagai pengawasan dan kontrol produk. Dari luasnya kemajuan teknologi, banyak industri atau perusahaan yang menginginkan pengaplikasian SCADA secara efisien dan mudah. Salah satu contohnya ialah pengaplikasian SCADA pada *smartphone* yang bersifat mudah dibawa kemana saja dan digunakan kapan saja. Penggunaan komputer pada pengaplikasian SCADA berbasis *mobile* bertujuan sebagai server utama pembuatan program yang dihubungkan ke unit proses, sedangkan *smartphone* digunakan sebagai kontrol serta monitoring proses yang dapat diakses melalui jaringan internet.

#### **2.2 Sistem Kontrol *Open Loop & Close Loop***

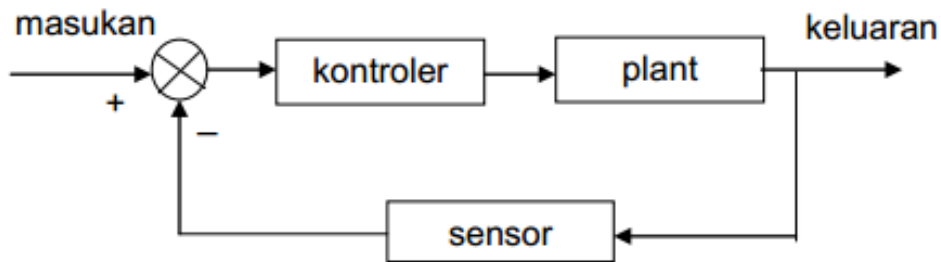
Sistem kontrol memiliki 2 jenis sistem yang biasa disebut dengan *open loop control* atau disebut juga kontrol lup terbuka dimana suatu sistem yang keluarannya tidak memiliki *feedback* atau umpan balik terhadap masukannya,



dan *close loop control* atau disebut sebagai kontrol lup tertutup dimana suatu sistem yang keluarannya memiliki *feedback* terhadap masukan yang berfungsi untuk memperkecil resiko kesalahan [4].



Gambar 2.1 *Open Loop Control* [4]



Gambar 2.2 *Close Loop Control* [4]

### 2.3 *Trainer kit*

*Trainer Kit* merupakan sebuah alat yang memiliki fungsi untuk mendukung atau menunjang suatu kegiatan pendidikan seperti praktikum guna mengembangkan SDM dalam menerapkan ilmu pengetahuan yang didapat dari benda. Penggunaan *trainer* dapat sebagai penunjang kegiatan selama belajar mengajar guna membuat mahasiswa memahami suatu konsep dalam praktikum yang didapat dari suatu benda nyata, serta dapat mengembangkan pola pikir mahasiswa [5].

### 2.4 IOT

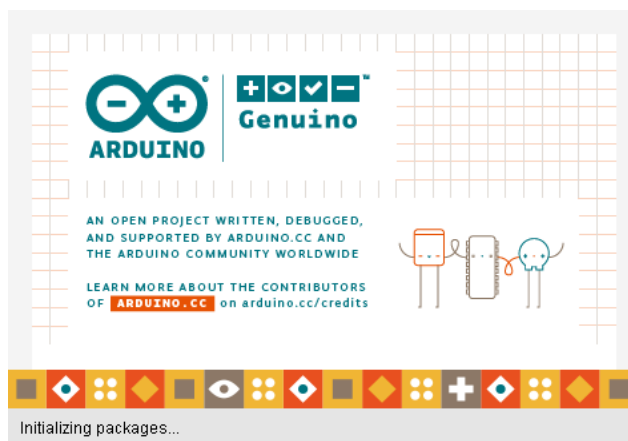
*Internet of Things*, atau yang biasa disebut dengan singkatan IOT merupakan sebuah pemahaman yang memiliki tujuan untuk mengembangkan kegunaan dari luasnya cakupan internet yang cukup maju serta terhubung tanpa

henti setiap harinya seperti data. IoT memiliki ciri khas atau kemampuan untuk transfer data lewat jaringan tanpa bantuan perangkat lain seperti komputer serta manusia. Melalui algoritma Bahasa pemrograman yang telah disusun, maka akan membentuk sebuah sinyal yang membantu perangkat keras untuk mengeksekusi suatu perintah sehingga tidak diperlukan kendali dari manusia sehingga tidak dikontrol secara manual namun otomatis [2].

Salah satu Metode pada IOT adalah nirkabel atau pengendalian secara otomatis tanpa kabel melainkan melalui jaringan tanpa adanya jarak baik jauh maupun dekat berbasis sensor. Penerapan IoT dapat dikatakan luas karena sudah terdapat pada beberapa lingkungan kehidupan sekitar seperti contohnya *smart home* yang mana kita dapat melakukan pengontrolan lampu, menutup pintu, menyalakan televisi sebagai penghematan energi.

## 2.5 Arduino IDE

Arduino IDE atau singkatan dari *Integrated Development Environment* adalah suatu platform yang digunakan untuk kegiatan *programming* baik itu membuat, atau mengedit program pada arduino atau dengan kata lain memprogram board pada arduino [6].

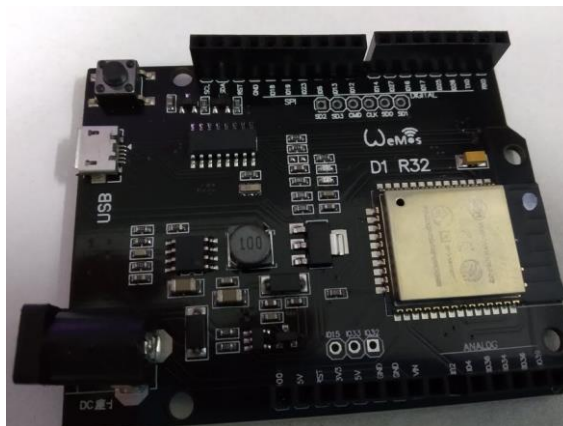


Gambar 2.3 Arduino IDE

## 2.6 ESP32

ESP32 merupakan salah satu jenis *controller* yang dapat mengakses jaringan *wifi* dan *bluetooth* yang dapat mendukung penggunaan IoT, dengan jumlah pin sebanyak 32 pin yang terdiri dari pin analog, pin digital, *source*, *ground*, serta SPI. Biasanya, terdiri dari chip ESP32 yang dihubungkan dengan komponen lain yang berfungsi sebagai pengendali untuk rangkaian listrik [7].

Jenis *development board* yang digunakan adalah Modul Wemos D1-R32 dengan berbasis *wifi* dan *bluetooth* yang digunakan untuk untuk aplikasi IoT. Modul Wemos D1-R32 juga memiliki pwm pada setiap pin yang artinya semua pin dapat digunakan sebagai pwm.



Gambar 2.4 Modul Wemos D1-R32 [6]

## BAB III

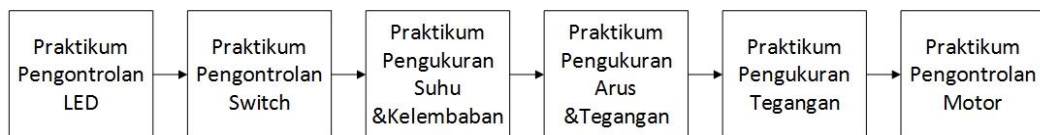
### METODE PELAKSANAAN

Pada bab ketiga ini, membahas tentang metode pelaksanaan yang dilakukan pada proyek akhir ini. Tahap-tahap yang dilakukan yakni sebagai berikut :

- a. Pengelompokan Jenis Praktikum
- b. Tahapan pelaksanaan
- c. Penggunaan Platform Blynk.

#### 3.1 Pengelompokan Jenis Praktikum

Tahap ini merupakan tahap awal dari pembuatan alat. Tahapan awal yang dilakukan ialah pengelompokan jenis praktikum penggunaan *trainer kit* yang akan mengacu pada beberapa praktikum seperti pada Gambar 3.1 dibawah ini.

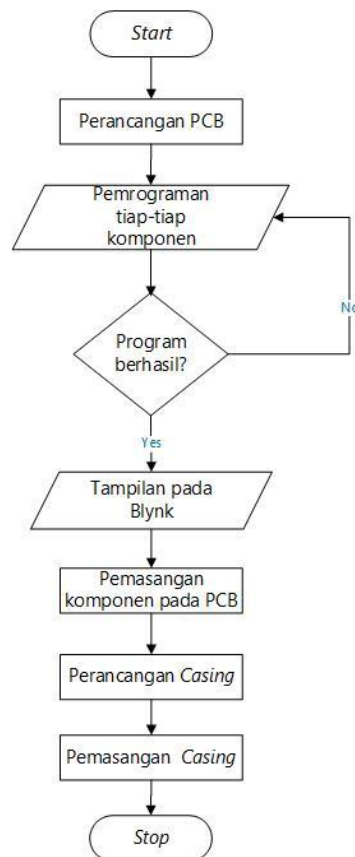


Gambar 3.1 Blok Diagram *Trainer Kit*

Berdasarkan Gambar 3.1 diatas, terdapat 6 praktikum. Setiap praktikum dilakukan secara berurutan dan tersusun dimulai dari praktikum LED hingga praktikum pengontrolan *driver* motor & motor. Praktikum yang dilakukan secara berurutan dilakukan guna memudahkan dalam proses pembuatan *trainer kit* serta memudahkan mahasiswa dalam mengelompokan data hasil pada tiap-tiap praktikum.

### 3.2 Tahapan Pelaksanaan

Yang dilakukan selanjutnya ialah membuat tahapan pelaksanaan dari pembuatan *trainer kit control system and data acquisition*. Dibawah ini merupakan gambar flowchart tahapan pelaksanaan pembuatan *trainer kit*.



Gambar 3.2 Flowchart Tahapan Pelaksanaan

### 3.3 Penggunaan Platform Blynk

Untuk melakukan penampilan nilai data, dan pengontrolan setiap komponen, maka pada tahap ini dilakukan melalui platform blynk. Blynk merupakan platform yang dapat digunakan pada *smartphone* iOS maupun android untuk pengendalian modul Arduino, modul Wemos D1-R32, serta modul sejenis melalui jaringan internet [8]. Platform blynk berperan sebagai tampilan dari program yang telah dibuat pada komputer menggunakan Arduino IDE. Komputer yang tersebut terhubung dengan modul wemos D1-R32 menggunakan media

komunikasi berupa kabel USB. *Widget* yang digunakan pada platform blynk juga dapat digunakan sebagai pengontrol beberapa komponen yang digunakan seperti LED, *switch* dan motor. Untuk penggunaan platform blynk pada tiap-tiap praktikum, langkah-langkah penggunaannya terdapat pada lampiran.

## BAB IV PEMBAHASAN

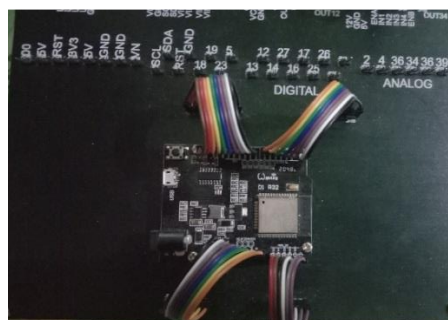
Pada bab ini, akan membahas secara detail mengenai proses pembuatan proyek akhir, dimulai dari penggunaan *software*, pemasangan komponen, tampilan pada blynk, serta pembuatan *casing*.

### 4.1 Uji Coba Tiap-Tiap Komponen

Uji coba dilakukan untuk mengecek kelayakan dari tiap-tiap komponen yang digunakan pada tiap-tiap praktikum. Selain itu, percobaan komponen dilakukan untuk memudahkan dalam pembuatan rangkaian, serta melakukan *programming* didalamnya. Berikut beberapa komponen yang digunakan.

#### 4.1.1 Wemos D1-R32

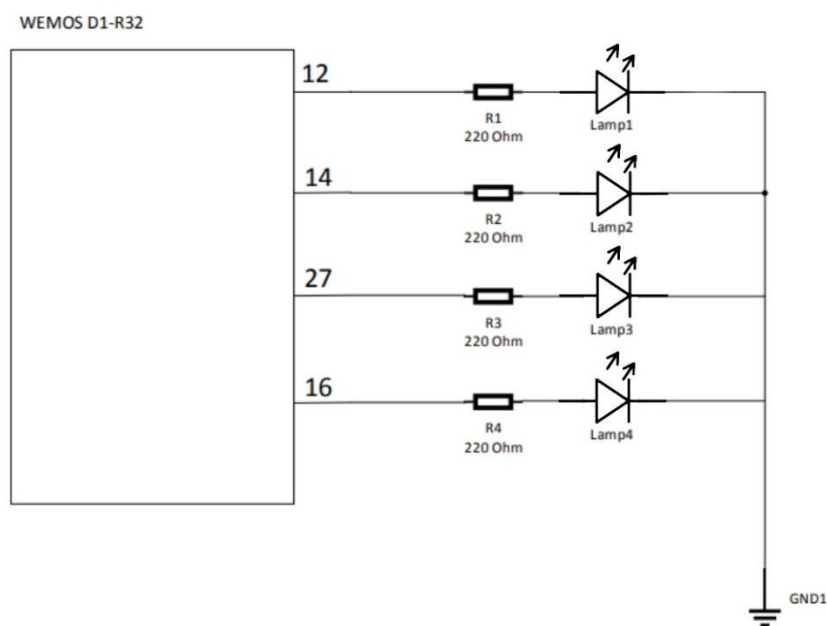
Komponen utama yang digunakan atau bisa disebut sebagai *controller* dalam pembuatan *trainer kit* ini adalah modul wemos D1-R32. Perakitan modul wemos D1-R32 pada papan PCB dilakukan pertama kali dengan menghubungkan pin yang ada pada modul wemos D1-R32 dengan pin *header* yang sudah dipasang pada papan PCB menggunakan kabel *jumper* jenis *male-female* sehingga memudahkan untuk menghubungkan tiap-tiap kaki komponen ke pin yang akan digunakan pada modul wemos D1-R32. Berikut ini gambar pemasangan modul wemos D1-R32 pada papan PCB.



Gambar 4.1 Pemasangan Modul Wemos D1-R32

#### 4.1.2 LED

Komponen selanjutnya yang digunakan ialah LED (*light emitting diode*) yang digunakan sebagai lampu indikator kontrol nyala dan mati pada praktikum pengontrolan LED. Menggunakan resistor sebesar 220 ohm, serta lampu LED berwarna merah, biru dan hijau.



Gambar 4.2 Gambar Rangkaian Pengontrolan LED

Gambar diatas digunakan sebagai acuan untuk melakukan praktikum pengontrolan LED. Pengujian pada LED sebelumnya dilakukan menggunakan breadboard. Setelah pengujian berhasil dilakukan menggunakan program, maka setelah itu dilakukan pemasangan komponen LED serta resistor ke papan PCB yang sebelumnya sudah dirancang terlebih dahulu. Dibawah ini merupakan gambar dari pemasangan LED pada papan PCB.





Gambar 4.3 Pemasangan LED Pada Papan PCB

Program yang digunakan pada praktikum pengontrolan LED adalah program yang dapat langsung mengontrol nyala/mati LED melalui platform blynk. Penggunaan aplikasi blynk untuk praktikum pengontrolan LED sudah terlampir pada lampiran serta modul praktikum. Berikut penjelasan dari program yang digunakan.

```

WidgetLED led1(V26);
WidgetLED led2(V27);
WidgetLED led3(V28);
WidgetLED led4(V29);
}
Program untuk tampilan pada
platform blynk.

char auth[] = "FGQlb9yzb4qbO2cFNMRALEtR3b1UYHxo";
char ssid[] = "cicak kabur";
char pass[] = "apasihkamu";
}
Program untuk
mengakses blynk
melalui jaringan
internet, dengan
memasukkan id dan
password internet .

int LEDPin = 18;
int LEDPin2 = 19;
int LEDPin3 = 23;
int LEDPin4 = 5;
}
Memasukkan pin input LED yang
digunakan pada modul WEMOS.

void setup(){
  Serial.begin(115200);
}
  
```

Tampilan pada *serial monitor*.

`Blynk.begin(auth, ssid, pass)` → Menjalankan program dan menghubungkan pada blynk melalui jaringan internet setelah memasukkan token, *id*, dan *password*.

```
pinMode (LEDPin, OUTPUT);  
pinMode (LEDPin2, OUTPUT);  
pinMode (LEDPin3, OUTPUT);  
pinMode (LEDPin4, OUTPUT);  
}
```

Mengakses LED sebagai *output* melalui `pinMode`.

```
BLYNK_WRITE(V21) {  
  int pinValue = param.asInt();  
  if (pinValue == 1) {  
    digitalWrite(LED1, HIGH);  
    led1.on();  
  }
```

Program untuk menhidupkan LED 1.

```
  else {  
    digitalWrite(LED1, LOW);  
    led1.off();  
  }  
}
```

Program untuk mematikan LED 1.

```
BLYNK_WRITE(V22) {  
  int pinValue = param.asInt();  
  if (pinValue == 1) {  
    digitalWrite(LED2, HIGH);  
    led2.on();  
  }
```

Program untuk menhidupkan LED 2.

```
  else {  
    digitalWrite(LED2, LOW);  
    led2.off();  
  }  
}
```

Program untuk mematikan LED 2.

```
BLYNK_WRITE(V23) {  
  int pinValue = param.asInt();  
  if (pinValue == 1) {  
    digitalWrite(LED3, HIGH);  
    led3.on();  
  }
```

Program untuk menhidupkan LED 3.

```

}
else {
  digitalWrite(LED3, LOW);
  led3.off();
}
}
BLYNK_WRITE(V24) {
  int pinValue = param.asInt();
  digitalWrite(LED4, HIGH);
  led4.on();
}
else {
  digitalWrite(LED4, LOW);
  led4.off();
}
}
void loop()
{
  Blynk.run();
}

```

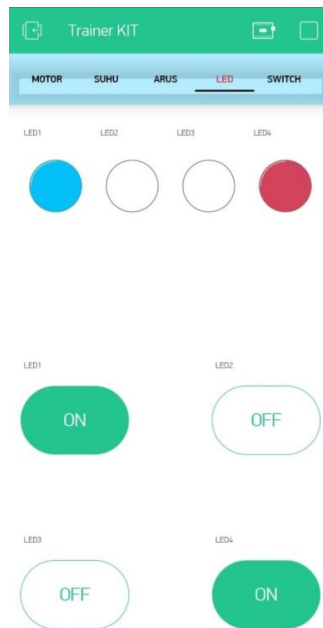
Program untuk mematikan LED 3.

Program untuk menhidupkan LED 4.

Program untuk mematikan LED 4.

Program untuk menjalankan program yang sudah dibuat dan menampilkan pada blynk.

Program diatas, akan menghasilkan keluaran berupa data nyala/mati dari tiap-tiap lampu yang dihasilkan dari pengontrolan melalui platform blynk. Pengontrolan nyala mati LED dilakukan dengan menekan tombol *On* pada layar *smartphone* maka LED akan menyala, dan dengan menekan tombol *Off* pada layar *smartphone* maka LED akan mati.

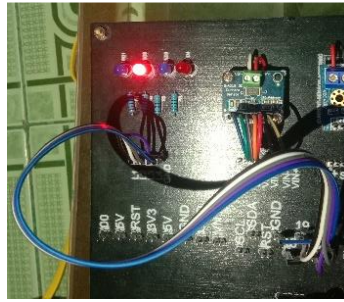


Gambar 4.4 Tampilan Desain LED Blynk

Untuk lebih jelasnya lampu akan menyala dan mati seperti gambar dibawah ini pada saat masing-masing LED menyala yang dikontrol melalui blynk. Saat tombol *on* LED1 ditekan, maka akan menyebabkan lampu LED1 menyala. Ketika tombol *off* LED1 ditekan, maka lampu LED1 akan mati. Hal yang sama terjadi pada LED lainnya.



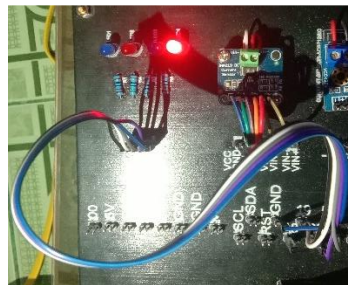
Gambar 4.5 Lampu 1 Menyala



Gambar 4.6 Lampu 2 Menyala



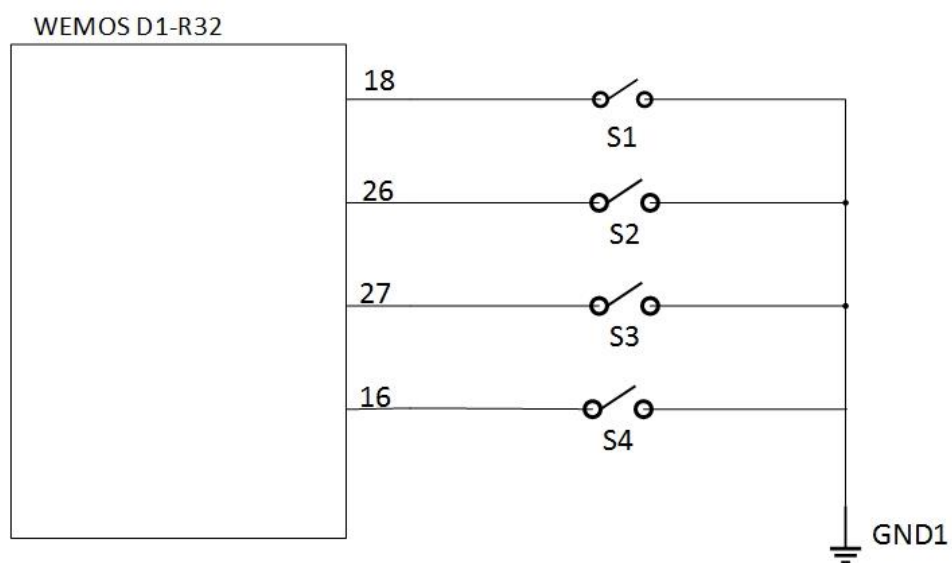
Gambar 4.7 Lampu 3 Menyala



Gambar 4.8 Lampu 4 Menyala

### 4.1.3 Switch

Komponen selanjutnya yang digunakan ialah *switch*, berfungsi sebagai penghubung serta pemutus suatu rangkaian. Seperti gambar dibawah ini, rangkaian *switch* digunakan untuk mengidentifikasi *on/off* pada *switch* melalui *monitoring* blynk pada praktikum pengontrolan *switch*. Pada rangkaian dibawah ini, outputnya menggunakan logika *high*.



Gambar 4.9 Rangkaian Switch

Gambar diatas digunakan sebagai acuan untuk melakukan praktikum pengontrolan *switch*. Setelah pengujian berhasil dilakukan menggunakan program, maka setelah itu dilakukan pemasangan komponen *switch* ke papan PCB yang sebelumnya sudah dirancang terlebih dahulu. Dibawah ini merupakan gambar dari pemasangan *switch*.



Gambar 4.10 Pemasangan *Switch*

Program yang digunakan pada praktikum pengontrolan *switch* adalah program yang dapat langsung mengontrol *switch* melalui platform blynk. Penggunaan aplikasi blynk untuk praktikum pengontrolan *switch* sudah terlampir pada lampiran serta modul praktikum . Berikut penjelasan dari program yang digunakan.

WidgetLCD lcd (V11); → Penggunaan pin *virtual* 11 untuk tampilan pada platform blynk.

const int pinButton = 25; → Penggunaan pin 25 pada *controller* untuk akses *Switch*.

char auth[] = "dJi22\_5iTKPZTWJhGASelxMcQ-ntnp0G";  
char ssid[] = "cicakkabur";  
char pass[] = "apasihkamu";

} Program untuk mengakses blynk melalui jaringan internet, dengan memasukkan *id* dan *password* internet .

void setup() {  
  Blynk.begin(auth, ssid, pass); → Menjalankan program pada blynk.

pinMode(pinButton, INPUT\_PULLUP); → Program untuk mengakses *switch* sebagai *input* .  
}

void loop() {  
  Blynk.run();

} Mengeksekusi program yang telah dibuat sebelumnya, dan menjalankannya pada Blynk.

```

if(digitalRead(pinButton) == LOW){
  lcd.clear();
  lcd.print(0,0,"KONDISI SAAT INI");
  lcd.print(0,1,">>SWITCH ON");
}

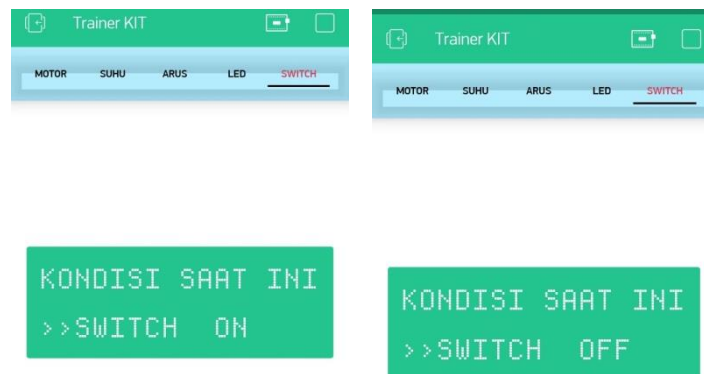
else{
  lcd.print(0,0,"KONDISI SAAT INI");
  lcd.print(0,1,">>SWITCH OFF");
}

```

Program untuk memberikan keterangan *switch on* pada tampilan blynk.

Program kebalikan pada *if* maka keterangan *switch off* akan muncul pada tampilan blynk.

Program diatas, akan menghasilkan keluaran berupa data kondisi *on/off* dari pengontrolan melalui platform blynk. Pengontrolan *on/off* dilakukan pada *switch* dan data tampilan *on/off* akan tampil pada platform blynk seperti gambar dibawah ini.



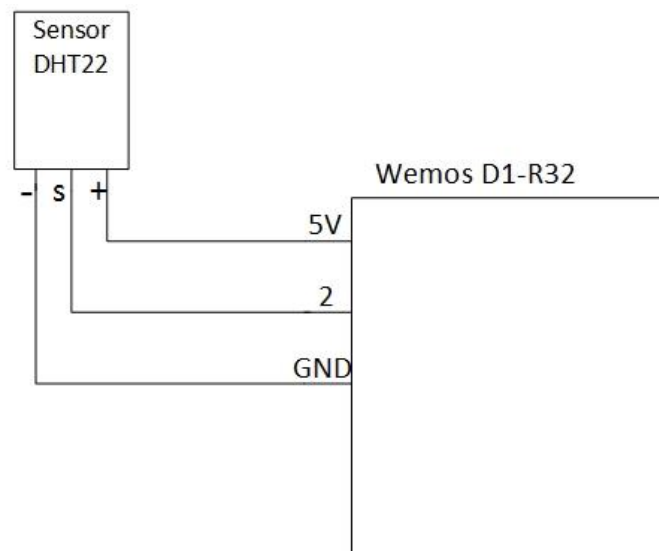
Gambar 4.11 Tampilan Blynk Pengontrolan *Switch*

Gambar diatas menunjukkan tampilan pada platform blynk pada saat kondisi *switch* ketika *on/off*.



#### 4.1.4 Sensor DHT22

Komponen yang digunakan selanjutnya adalah sensor DHT22 yang mana berfungsi sebagai pengukur suhu dan kelembaban udara yang ada disekitarnya. Pengambilan nilai suhu dan kelembaban ini dilakukan pada sebuah ruangan yang memiliki banyak ventilasi udara. Dan berikut adalah rangkaian dari sensor DHT22.



Gambar 4.12 Rangkaian Sensor Suhu

Gambar diatas digunakan sebagai acuan untuk melakukan praktikum pengukuran suhu dan kelembaban pada sekitarnya. Setelah pengujian berhasil dilakukan menggunakan program, maka setelah itu dilakukan pemasangan komponen sensor DHT22 ke papan PCB yang sebelumnya sudah dirancang terlebih dahulu. Dibawah ini merupakan gambar dari pemasangan sensor DHT22 pada papan PCB.



Gambar 4.13 Pemasangan Sensor DHT22 Pada Papan PCB

Program yang digunakan pada praktikum pengukuran suhu dan kelembaban adalah program yang dapat langsung menampilkan nilai *temperature* dan *humidity* serta grafiknya melalui platform blynk. Penggunaan aplikasi blynk untuk praktikum pengukuran suhu & kelembaban sudah terlampir pada lampiran serta modul praktikum . Berikut penjelasan dari program yang digunakan.

```

char auth[] = "Te_GwKqUuhakioTauPrAVWeTSIOTo9wA";
char ssid[] = "cicakkabur";
char pass[] = "apasihkamu";

void setup() {
  Blynk.begin(auth, ssid, pass);

  Serial.begin(9600);
  Serial.println("Deteksi Suhu dan kelembaban");
  dht.begin();
}

void loop() {
  delay(1000);

  float t = dht.readTemperature();
  float h = dht.readHumidity();

  if (isnan(h) || isnan(t)) {
    Serial.println("Sensor tidak terbaca!");
    return;
  }

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.println(" *C ");

```

Program untuk mengakses blynk melalui jaringan internet, dengan memasukkan *id* dan *password*.

Menjalankan program dan menghubungkan pada blynk melalui jaringan internet setelah memasukkan token, *id*, dan *password*.

Mendeteksi suhu dan kelembaban .

Menjalankan program yang sudah dibuat sebelumnya, dengan waktu jeda 1 detik.

Pembacaan variabel atau nilai suhu dan kelembaban dengan tipe angka desimal.

Program jika terjadi kesalahan baca atau sensor tidak berfungsi.

Tampilan pada *serial monitor* untuk “Humidity” dalam %. Dan “Temperature” dalam Celcius.

```

Blynk.virtualWrite(V4,t);
Blynk.virtualWrite(V5,h);
}

```

Program tampilan pada blynk. V4 pin virtual untuk tampilan nilai *temperature*, dan V5 pin virtual untuk tampilan *humidity*.

Program diatas akan menghasilkan keluaran berupa data nilai dan grafik dari *temperature* dan *humidity* pada tampilan platform blynk seperti gambar dibawah ini.



Gambar 4.14 Hasil Suhu & Kelembaban Pada Blynk

Pada gambar diatas dapat dilihat nilai *temperature* yang keluar bernilai 29,1 °C dan nilai *humidity* bernilai 73,8% juga bentuk grafik yang terlihat pada *temperature* dan *humidity* terlihat stabil.

Dari pengujian diatas, maka didapat nilai yang ditampilkan pada platform blynk dan nilai tersebut dimasukkan kedalam tabel dibawah ini dengan pengambilan nilai tiap 5 detik.

Tabel 4.1 Hasil Pengukuran Sensor DHT22

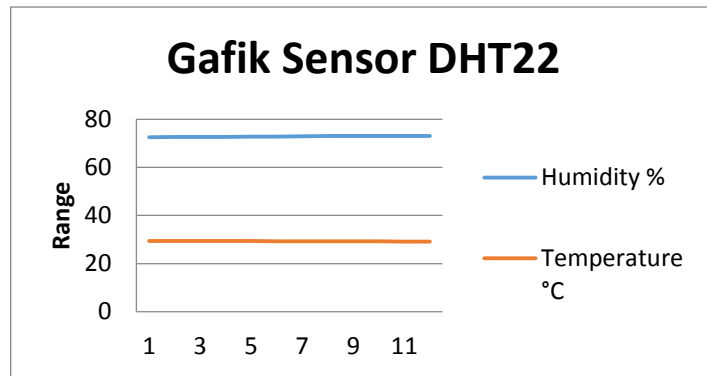
Waktu (s)	Sensor DHT22		Alat ukur		error %
	Humidity %	Temperature °C	Humidity %	Temperature °C	
5	72,50	29.4	71	28,6	0,02
10	72,60	29,40	71	28,6	0,02
15	72,60	29,40	71	28,6	0,02
20	72,70	29,40	71	28,6	0,02
25	72,80	29,40	71	28,6	0,02
30	72,80	29,30	71	28,6	0,02
35	72,90	29,30	71	28,7	0,03
40	73,10	29,30	72	28,6	0,01
45	73,10	29,30	72	28,6	0,01
50	73,10	29,30	72	28,6	0,01
55	73,10	29,20	72	28,6	0,01
60	73,10	29,20	72	28,6	0,01

Pada Table diatas, data diambil menggunakan sensor DHT22 dan alat ukur *temperature* dan *humidity* sehingga didapatkan perbandingan serta nilai *error* seperti pada tabel. Berikut ini gambar alat ukur yang digunakan.



Gambar 4.15 Pengukuran Suhu & Kelembaban Dengan Alat Ukur

Dari data Tabel tersebut memperlihatkan bahwa suhu dan kelembaban yang diukur oleh sensor DHT22 memiliki hasil yang sama dengan ketika mengukur menggunakan alat ukur suhu dan kelembaban dengan tingkat *error* sekitar 0,01

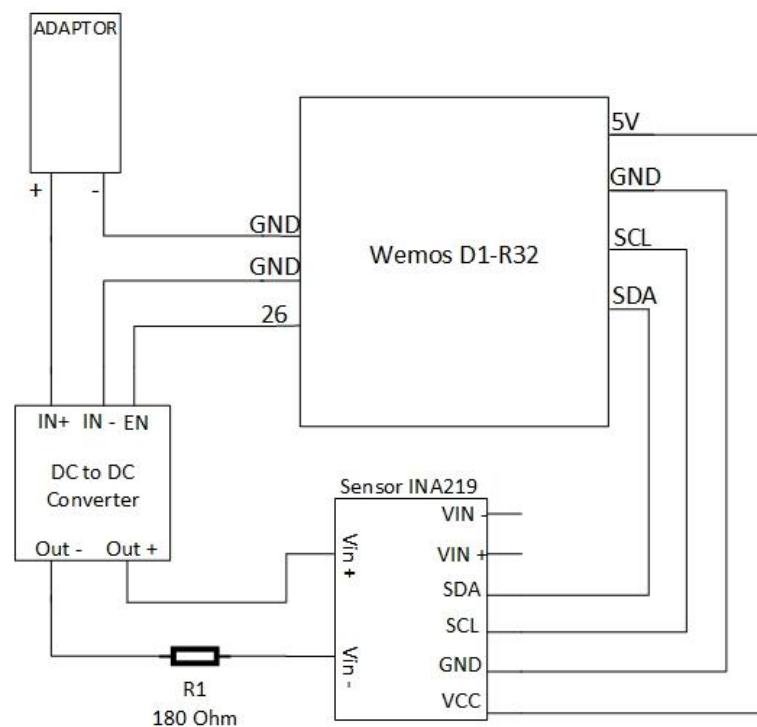


Gambar 4.16 Grafik Kestabilan Sensor DHT22

Gambar Grafik diatas menunjukkan pengukuran suhu dan kelembaban yang cukup stabil, menandakan bahwa sensor DHT 22 memiliki tingkat akurasi yang cukup baik dengan nilai *temperature* berada pada *range* 28 mendekati 30 °C, dan *humidity* berada pada *range* 71%.

#### 4.1.5 Sensor INA219

Komponen yang digunakan selanjutnya adalah sensor INA219 sebagai pengukur parameter arus serta tegangan pada praktikum pengukuran arus dan tegangan. Dengan menggunakan *adaptor supply* sebesar 12V/1A, dan beban resistor sebesar 180 Ohm. Sensor INA 219 juga menggunakan *buckboost* DC to DC Converter untuk mendapatkan nilai tegangan yang konstan, dan jenis yang digunakan pada rangkaian adalah jenis *boost* (penaik tegangan).



Gambar 4.17 Rangkaian Sensor INA219

Gambar diatas digunakan sebagai acuan untuk melakukan praktikum pengukuran tegangan dan arus. Setelah pengujian berhasil dilakukan menggunakan program, maka setelah itu dilakukan pemasangan komponen sensor INA219 ke papan PCB yang sebelumnya sudah dirancang terlebih dahulu. Dibawah ini merupakan gambar dari pemasangan sensor INA219.



Gambar 4.18 Pemasangan Sensor Arus INA219 Pada Papan PCB

Program yang digunakan pada praktikum pengukuran tegangan dan arus adalah program yang dapat langsung menampilkan nilai tegangan dan arus, serta grafiknya melalui platform blynk. Penggunaan aplikasi blynk untuk praktikum pengukuran tegangan dan arus sudah terlampir pada lampiran serta modul praktikum. Berikut penjelasan dari program yang digunakan.

```
char auth[] = "V7Jyybzk-oqwiRg2_4UhttlVdxY--1DZ";
char ssid[] = "cicakkabur";
char pass[] = "apasihkamu";
```

Program untuk mengakses blynk melalui jaringan internet, dengan memasukkan *id* dan *password*.

```
int enable1Pin = 26; → Setting Pin .
```

```
const int freq = 3000;
const int pwmChannel = 0;
const int resolution = 8;
```

Program pengaturan PWM.

```
void setup()
{
  Blynk.begin(auth, ssid, pass);
```

Menjalankan program dan menghubungkan pada blynk melalui jaringan internet setelah memasukkan token, *id*, dan *password*.

```
ledcSetup(pwmChannel, freq, resolution);
ledcAttachPin(enable1Pin, pwmChannel);
```

Program untuk menjalankan fungsi turun naik pwm pada aplikasi blynk.

```
BLYNK_WRITE(V2)
{
  int pinValue = param.asInt();
  ledcWrite(pwmChannel, pinValue);
  Blynk.virtualWrite(V1, pinValue);
  Serial.println(pinValue);
}
```

Program untuk menjalankan fungsi turun naik pwm pada aplikasi blynk.

```

void loop(void)
{
  Blynk.virtualWrite(V7,busvoltage);
  Blynk.virtualWrite(V10,current_mA);
  delay(2000);
}

```

Tampilan hasil pengukuran pada platform blynk.

Program diatas, akan menghasilkan keluaran berupa data nilai pengukuran tegangan dan arus yang ditampilkan pada platform blynk. Data yang keluar berupa nilai besarnya tegangan dan arus serta grafik seperti gambar dibawah ini.



Gambar 4.19 Tampilan Output Sensor Arus INA219

Dari pengujian sensor didapat nilai ukur pada tampilan blynk yang dimasukkan ke tabel dibawah ini.

Tabel 4.2 Hasil Nilai Tegangan dan Arus

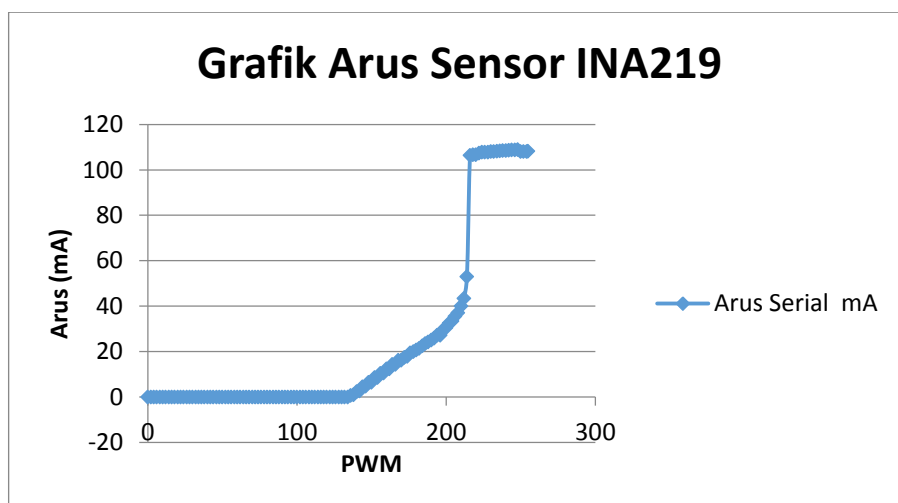
PWM	Tegangan pada Blynk (V)	Arus pada Blynk ( mA)
0-130	0	0
132	1	0
134	1	0
136	1.3	0.6



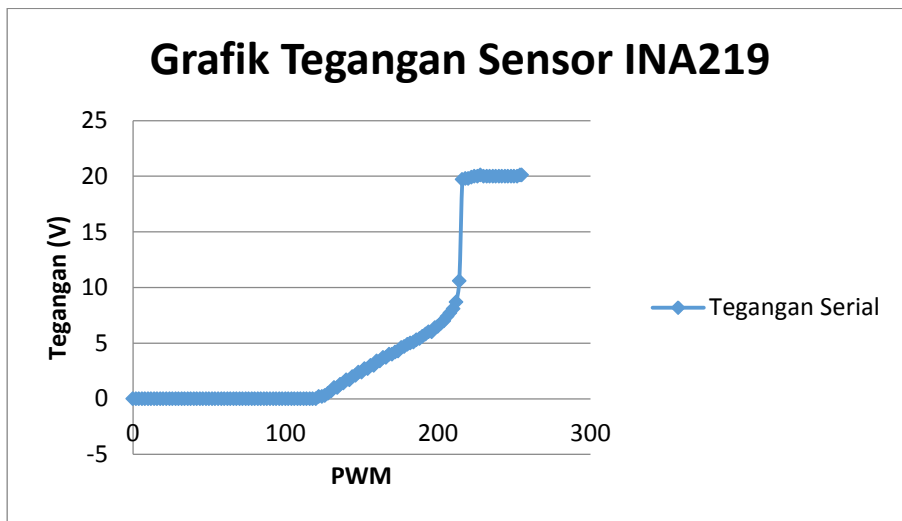
PWM	Tegangan pada Blynk (V)	Arus pada Blynk ( mA)
138	1.4	0.9
140	1.7	2.4
142	1.7	2.6
144	2	4.5
146	2.1	4.7
148	2.4	6.4
150	2.4	6.5
152	2.7	8.5
154	2.7	8.6
156	3	10.4
158	3	10.5
160	3.4	12.3
162	3.4	12.3
164	3.7	14.3
166	3.7	14.3
168	4	16.2
170	4	16.2
172	4.2	17.4
174	4.3	17.8
176	4.6	19.4
178	4.7	19.9
180	4.9	20.7
182	5	21.4
184	5.1	22.5
186	5.3	23.5
188	5.4	24.2
190	5.6	25.1
192	5.8	25.9
194	6	27.2
196	6	27.2
198	6.4	29.5
200	6.5	30.4
202	6.8	32.5
204	7	33.3
206	7.4	35.4
208	7.7	37
210	8.1	40
212	8.7	43.3

PWM	Tegangan pada Blynk (V)	Arus pada Blynk ( mA)
214	10.6	53
216	19.7	106.3
218	19.8	106.6
220	19.8	106.6
222	19.9	107.4
224	20	107.8
226	20	107.8
228	20.1	107.8
230	20	108
232	20	108.1
234	20	108.2
236	20	108.3
238	20	108.4
240	20	108.5
242	20	108.6
244	20	108.7
246	20	108.8
248	20	108.9
250	20	108.1
252	20	108.11
254	20.1	108.1
255	20.1	108.2

Dari data tabel diatas, didapat gambar grafik seperti dibawah ini



Gambar 4.20 Grafik Arus Sensor INA219

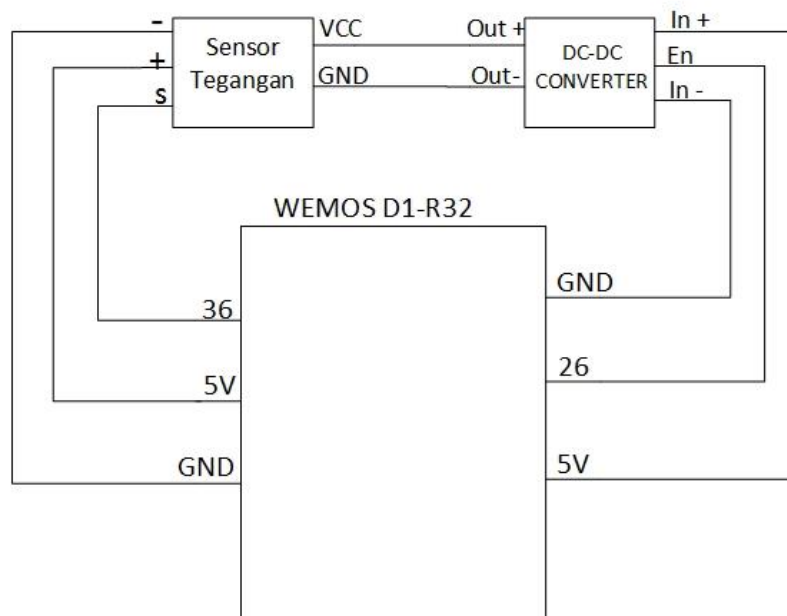


Gambar 4.21 Grafik Tegangan Sensor INA219

Grafik menunjukkan perubahan nilai tegangan dan arus yang diukur oleh sensor INA219 pada pwm 0 hingga  $\pm 130$  masih menunjukkan nilai 0, dan pada pwm 130 keatas hingga batas maksimum (255) menunjukkan nilai seiring dengan besarnya pwm yang digunakan.

#### 4.1.6 Sensor Tegangan

Komponen selanjutnya yang digunakan adalah sensor tegangan yang difungsikani untuk membaca nilai tegangan pada rangkaian melalui *controller* Wemos D1-R32 dari pin analog berupa pin 36. Dengan menggunakan komponen DC to DC *Converter* yang berfungsi untuk menaik dan menurunkan tegangan.



Gambar 4.22 Rangkaian Sensor Tegangan

Gambar diatas digunakan sebagai acuan untuk melakukan praktikum pengukuran tegangan pada suatu rangkaian. Setelah pengujian berhasil dilakukan menggunakan program, maka setelah itu dilakukan pemasangan komponen sensor tegangan ke papan PCB yang sebelumnya sudah dirancang terlebih dahulu. Dibawah ini merupakan gambar dari perakitan sensor tegangan.



Gambar 4.23 Pemasangan Sensor Tegangan Pada Papan PCB

Program yang digunakan pada praktikum pengukuran tegangan dan arus adalah program yang dapat langsung menampilkan nilai tegangan dan arus, serta grafiknya melalui platform blynk. Penggunaan aplikasi blynk untuk praktikum pengukuran tegangan sudah terlampir pada lampiran serta modul praktikum. Berikut penjelasan dari program yang digunakan.

```
char auth[] = "fk9oTwX5P8IHh7o4IFOSUd4IsT1I824D";
char ssid[] = "cicakkabur";
char pass[] = "apasihkamu";
```

Program untuk mengakses blynk melalui jaringan internet, dengan memasukkan *id* dan *password*.

```
const int voltage_sensor = 36;
float voltage_sensor_raw;
float v_beban;
```

Program untuk mengakses pin yang digunakan pada sensor tegangan berupa pin 36 dan mengakses nilai ukur tegangan berupa angka desimal.

```
int enable1Pin = 26;
const int freq = 3000;
```

Setting pin PWM.

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

Menjalankan program dan menghubungkan pada blynk melalui jaringan internet setelah memasukkan token, *id*, dan *password*.

```
ledcSetup(pwmChannel, freq, resolution);
ledcAttachPin(enable1Pin, pwmChannel);
```

Mengatur PWM.

```
pinMode (voltage_sensor,INPUT);
```

Konfigurasi fungsi pin.

BLYNK\_WRITE(V2){ → Penggunaan pin virtual pada blynk.

```
int pinValue = param.asInt();  
ledcWrite(pwmChannel, pinValue);  
Blynk.virtualWrite(V1, pinValue);  
Serial.println(pinValue);  
}
```

Program untuk menjalankan fungsi turun naik pwm pada aplikasi blynk.

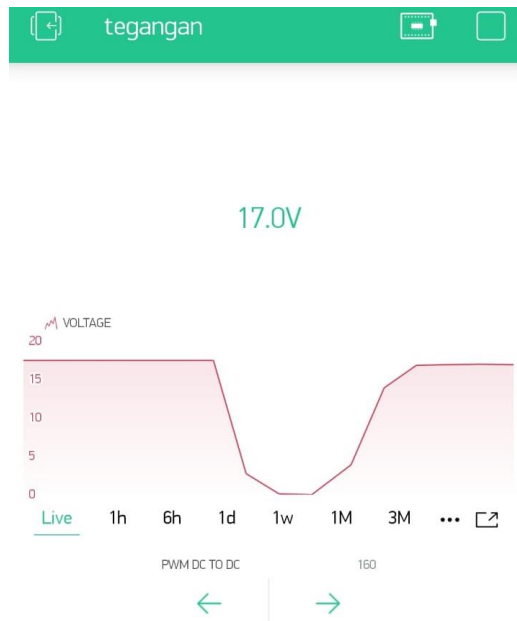
```
void loop(void)  
{  
  Blynk.run();  
  baca_sensor();  
  Serial.print ("NILAI TEGANGAN:");  
  Serial.println (v_beban);  
  Blynk.virtualWrite(V7,v_beban);  
  delay(2000);  
}
```

Menjalankan program yang sudah dibuat untuk membaca nilai sensor dan ditampilkan pada platform blynk.

```
void baca_sensor()  
{  
  voltage_sensor_raw = analogRead(voltage_sensor);  
  v_beban = map(voltage_sensor_raw,0,1023,0,2500);  
  v_beban = v_beban/100*0.175;  
}
```

Program pembacaan sensor tegangan.

Program diatas menghasilkan keluaran berupa data nilai tegangan dan grafik pada tampilan platform blynk seperti gambar dibawah ini. Pengukuran dilakukan menggunakan DC to DC *Converter* dengan tegangan yang diatur pada 25V.



Gambar 4.24 Hasil Sensor Tegangan

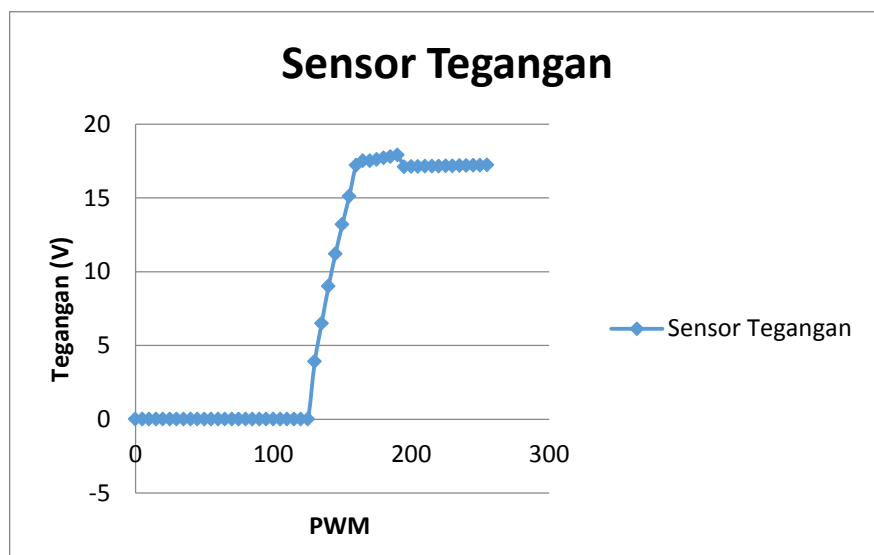
Dari contoh tampilan blynk diatas, terlihat nilai tegangan dan grafik yang ditampilkan pada platform blynk bernilai 17 V pada pwm 160. Nilai yang keluar kemudian dimasukkan kedalam tabel dibawah ini.

Tabel 4.3 Tabel hasil pengujian sensor tegangan

PWM	Sensor Tegangan	Multimeter
0-125	0	0
130	3.9	4.4
135	6.5	6.9
140	9	9.7
145	11.2	11.5
150	13.2	13.4
155	15.1	14.6
160	17.2	15.5
165	17.5	16
170	17.5	16.4
175	17.6	16.6
180	17.7	16.8

PWM	Sensor Tegangan	Multimeter
185	17.8	17
190	17.9	17.2
195	17.1	17.5
200	17.11	17.5
205	17.12	17.6
210	17.13	17.8
215	17.14	18.2
220	17.15	20
225	17.16	20
230	17.17	20
235	17.18	20
240	17.19	20
245	17.2	20
250	17.21	20
255	17.22	20

Dari data tabel diatas maka didapat bentuk grafik seperti gambar dibawah ini.



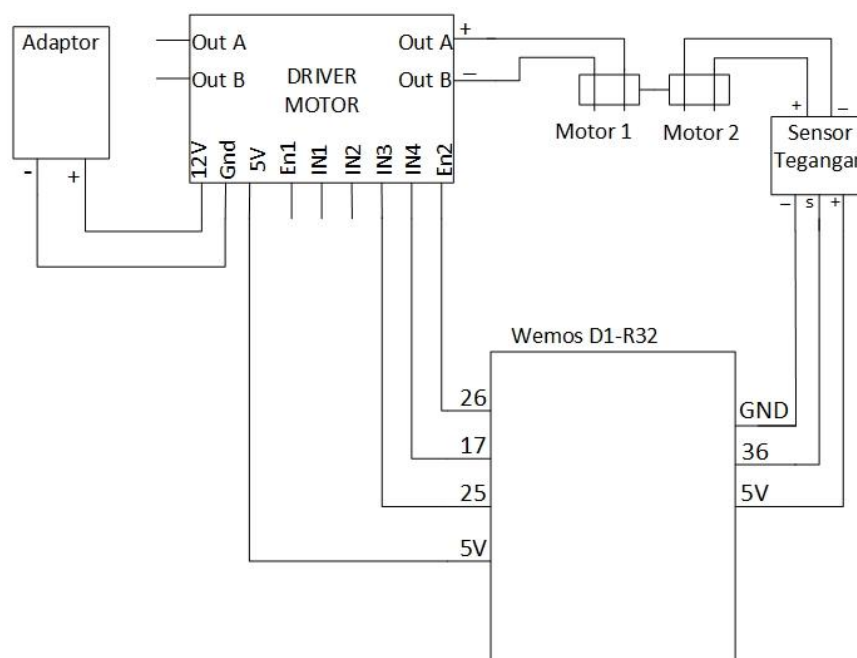
Gambar 4.25 Grafik Hasil Sensor Tegangan

Pada gambar grafik diatas dapat terlihat bahwa sensor tegangan dapat mengukur atau memberikan nilai keluaran pada pwm 130 hingga batas maksimum (255).



#### 4.1.7 Driver Motor dan Motor

Komponen terakhir yang digunakan ialah *driver* motor tipe L298n dan 2 buah motor yang disatukan menggunakan kopling. Penggunaan kedua motor ini memiliki fungsi masing-masing. Pada motor DC 1 motor digunakan sebagai indikasi pergerakan CW dan CCW. Sedangkan pada motor DC 2 digunakan sebagai pengukur tegangan yang dihubungkan langsung ke sensor tegangan untuk identifikasi tegangan yang keluar saat motor berputar.



Gambar 4.26 Rangkaian *Driver* Motor & Motor

Gambar diatas digunakan sebagai acuan untuk melakukan praktikum pengontrolan motor. Setelah pengujian berhasil dilakukan menggunakan program, maka setelah itu dilakukan perakitan komponen *driver* motor ke papan PCB dan pemasangan motor pada akrilik yang sebelumnya sudah dirancang terlebih dahulu. Dibawah ini merupakan gambar dari pemasangan *driver* motor pada papan PCB dan motor pada akrilik.



Gambar 4.27 Pemasangan *Driver Motor & Motor*

Program yang digunakan pada praktikum pengontrolan motor adalah program yang dapat langsung menampilkan nilai serta mengatur motor DC melalui platform blynk. Penggunaan aplikasi blynk untuk praktikum pengontrolan motor sudah terlampir pada lampiran serta modul praktikum. Berikut penjelasan dari program yang digunakan.

```

char auth[] = "dJi22_5iTKPZTWJhGASelxMcQ-ntnp0G";
char ssid[] = "cicakkabur";
char pass[] = "apasihkamu";

// sensor tegangan
const int voltage_sensor = 36;
float voltage_sensor_raw;
float v_beban;

int MOTORA = 17;
int MOTORB = 25;
int enable1Pin = 26;

const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 170;

WidgetLCD lcd(V1);

```

Program untuk mengakses blynk melalui jaringan internet.

Program untuk mendeteksi tegangan selama motor berputar.

Mengidentifikasi penggunaan pin untuk motor 1 dan *enable*.

Pengaturan PWM .

V1 digunakan sebagai tampilan LCD pada blynk.

```
void setup()
{
  //sensor tegangan
  pinMode (voltage_sensor,INPUT);
  Serial.begin(115200);
  pinMode(MOTORA, OUTPUT);
  pinMode(MOTORB, OUTPUT);
}

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

Pemanggilan fungsi pinmode sensor tegangan pada *void setup* sebagai *input*.  
 Konfigurasi motor sebagai *output*.  
 Menjalankan program dan menghubungkan pada blynk.

ledcSetup(pwmChannel, freq, resolution); → Menjalankan fungsi PWM.

ledcAttachPin(enable1Pin, pwmChannel); → Menjalankan fungsi enable pada *driver* motor, dan menjalankan fungsi PWM.

```
BLYNK_WRITE(V2)
{
  int pinValue = param.asInt();
  ledcWrite(pwmChannel, pinValue);
  Blynk.virtualWrite(V1, pinValue);
  Serial.println(pinValue);
}
```

Program untuk menjalankan fungsi turun naik pwm pada *enable driver* motor pada aplikasi blynk.

```
BLYNK_WRITE(V3) {
  int pinValue = param.asInt();
  if (pinValue == 1) {
    digitalWrite(MOTORA, HIGH);
    digitalWrite(MOTORB, LOW);
    ledcWrite(pwmChannel, dutyCycle);
    dutyCycle = dutyCycle;
    lcd.clear();
    lcd.print(0,0,"Berputar CCW");
  }
}
```

Program fungsi tombol untuk menjalankan motor secara ccw saat *widjet* pada aplikasi blynk ditekan. Penambahan PWM akan membuat kecepatan putar pada motor berubah secara teratur dari 170 hingga 255.

```
else {
  digitalWrite(MOTORA, LOW);
  digitalWrite(MOTORB, LOW);
  lcd.clear();
  lcd.print(0,0,"Motor Stop");
}
```

Program *else* berfungsi jika kondisi *if* tidak terpenuhi, maka motor kan berhenti bergerak dan menampilkan pada monitor blynk "Motor Stop".

```

BLYNK_WRITE(V0) {
  int pinValue = param.asInt();
  if (pinValue == 1) {
    digitalWrite(MOTORA, LOW);
    digitalWrite(MOTORB, HIGH);
    ledcWrite(pwmChannel, dutyCycle);
    dutyCycle = dutyCycle;
    lcd.clear();
    lcd.print(0,0,"Berputar CW");
  }

```

Program untuk menjalankan fungsi tombol motor secara CW saat *widget* pada aplikasi blynk ditekan. Penambahan PWM akan membuat kecepatan putar pada motor berubah secara teratur dari 170 hingga 255.

```

else {
  digitalWrite(MOTORA, LOW);
  digitalWrite(MOTORB, LOW);
  lcd.clear();
  lcd.print(0,0,"Motor Stop");

```

Program *else* berfungsi jika kondisi *if* tidak terpenuhi, maka motor kan berhenti bergerak dan menampilkan pada monitor blynk "Motor Stop".

```

void loop()
{
  Blynk.run();
  Blynk.virtualWrite(V6,v_beban);
  delay(3000);
}

```

Program eksekusi dari program yang sudah dibuat seperti membaca tegangan dan ditampilkan pada blynk.

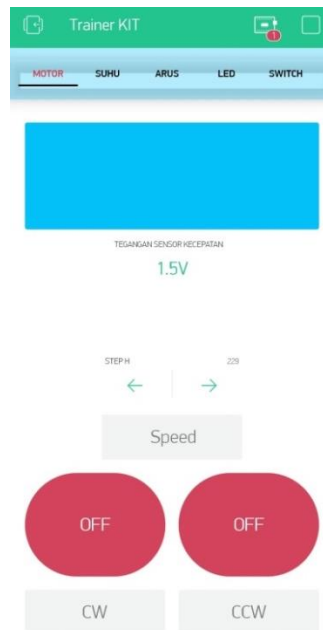
```

void baca_sensor()
{
  voltage_sensor_raw = analogRead(voltage_sensor);
  v_beban = map(voltage_sensor_raw,0,1023,0,2500);
  v_beban = v_beban/100/5.6;
}

```

Program untuk membaca tegangan yang keluar pada motor melalui sensor tegangan.

Program diatas akan menghasilkan keluaran data berupa putaran motor secara CW dan CCW dan besarnya tegangan yang keluar saat motor berputar pada tampilan platform blynk seperti gambar dibawah ini.



Gambar 4.28 Tampilan Pada Blynk

Gambar diatas menunjukkan pin *virtual* yang digunakan pada blynk yaitu untuk menampilkan kata “Motor Stop” pada LCD berwarna biru, lalu untuk menampilkan tegangan sensor kecepatan yang menggunakan motor DC 2 sebesar 1,5 V, lalu untuk tampilan PWM yang diatur, lalu tampilan CW dan CCW.

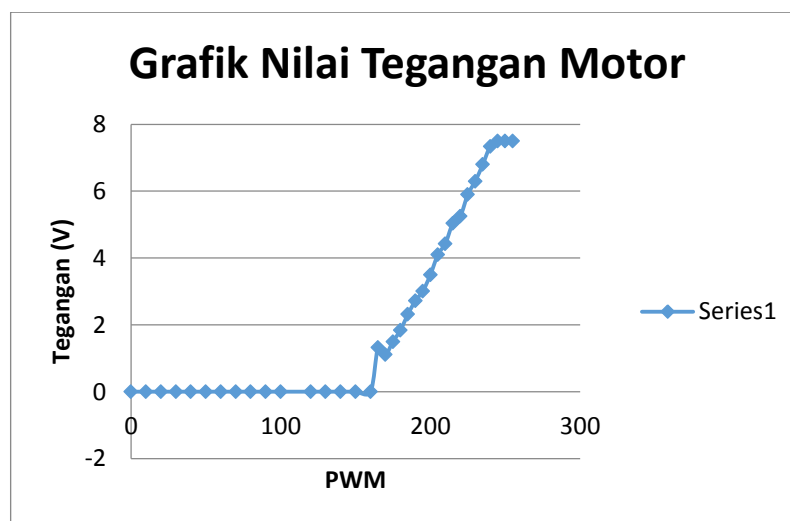
Setelah melakukan pengujian, dibawah ini merupakan tabel hasil pengukuran tegangan yang keluar selama motor berputar.

Tabel 4.4 Tabel Kecepatan Motor

PWM	Pengukuran Multimeter	Pengukuran sensor tegangan
0-160	0	0
165	1,32	0,96
170	1,11	1,0
175	1,50	1,10
180	1,84	1,43
185	2,32	1,68
190	2,72	1,87
195	3,01	2,77
200	3,50	3,05
205	4,10	4,25
210	4,43	4,27

PWM	pengukuran Multimeter	pengukuran sensor tegangan
215	5,04	5,50
220	5,25	5,79
225	5,90	6,31
230	6,30	6,54
235	6,80	7,20
240	7,34	7,61
245	7,5	7,9
250	7,5	7,9
255	7,5	8,0

Dari data tabel diatas, didapatkan gambar grafik nilai tegangan yang keluar saat motor berputar seperti dibawah ini

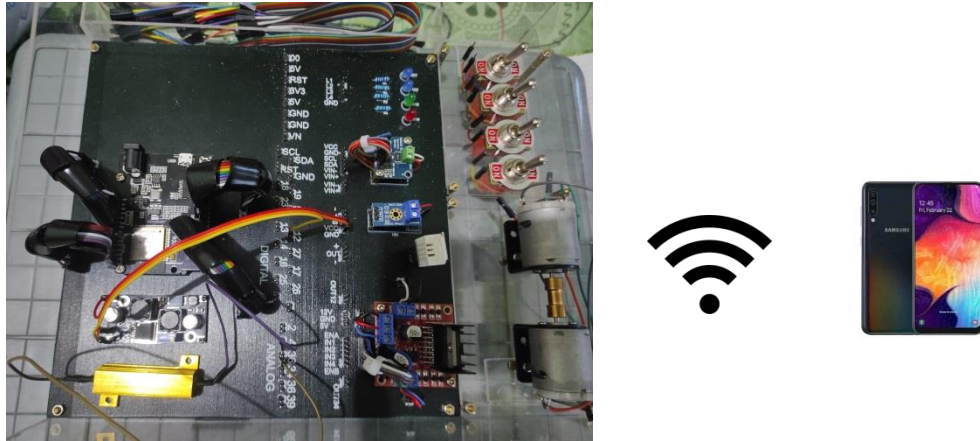


Gambar 4.29 Grafik Tegangan Motor

## 4.2 Desain Trainer Kit

Setelah tiap-tiap komponen diuji dan berhasil menampilkan keluaran, lalu pemasangan pada PCB dan akrilik sudah dilakukan, kegiatan selanjutnya adalah membuat desain *trainer kit*. Desain *trainer kit* seperti dibawah ini dibuat sesuai dengan kebutuhan pembelajaran. Semua komponen disusun sedemikian rupa untuk memuat tiap-tiap komponen dan elemen penunjang lainnya seperti kabel jumper sebagai penghubung kaki-kaki komponen.

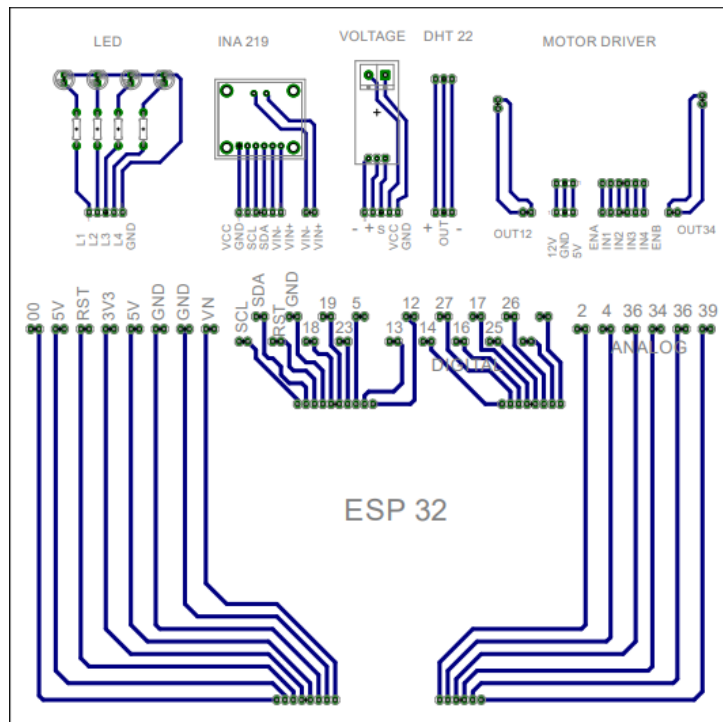
Penggunaan laptop hanya sebagai penghubung program dari aplikasi Arduino IDE ke *controller* menggunakan kabel usb dan selebihnya menggunakan jaringan *wi-fi*, maka pengontrolan dan pengambilan data dapat dilakukan menggunakan *smartphone* melalui aplikasi Blynk.



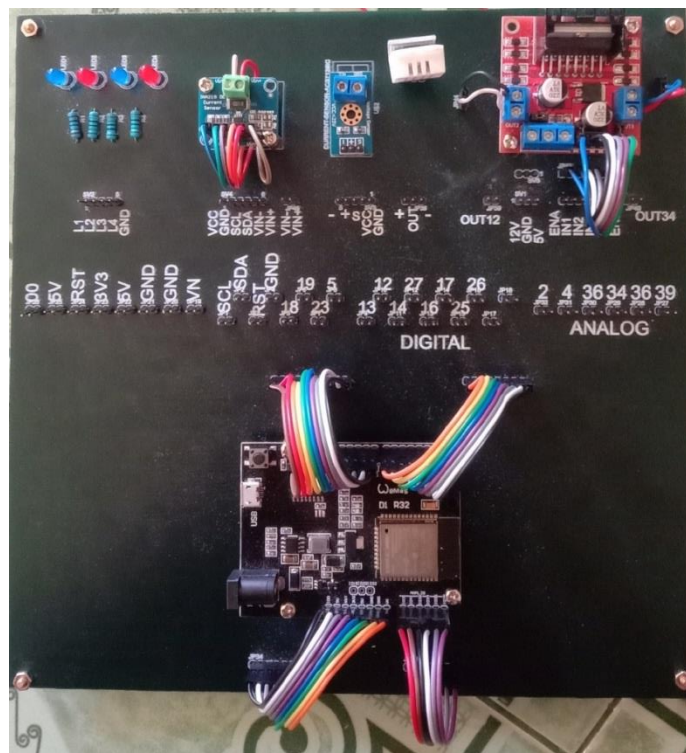
Gambar 4.30 Desain Trainer Kit

### 4.3 Layout PCB Secara Keseluruhan Komponen

Perancangan layout PCB difungsikan untuk menentukan posisi dari tiap-tiap komponen yang ada didalam trainer kit serta menghubungkan antara satu komponen dengan komponen lainnya melalui circuit. Berikut dibawah ini gambar dari layout PCB yang dibuat menggunakan aplikasi Eagle.



Gambar 4.31 Desain Layout PCB 2D

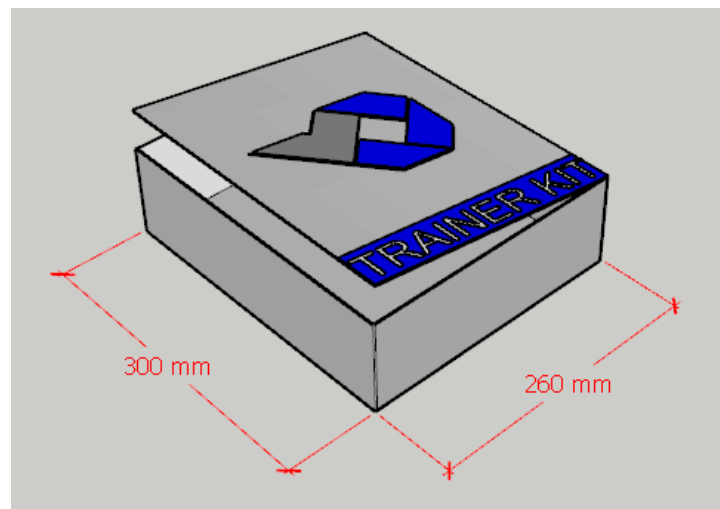


Gambar 4.32 Hasil Pemasangan Secara Keseluruhan



#### 4.4 Perakitan *Casing*

Pada tahap ini perakitan *casing* dilakukan untuk memberikan perlindungan pada komponen didalamnya dan mudah untuk dipindahkan dari satu tempat ke tempat lainnya. *Casing* terbuat dari bahan akrilik dengan ketebalan 2 mm. Berikut adalah gambar serta ukuran dari *casing*



Gambar 4.33 Desain *Casing*

#### 4.5 Kuisisioner

Kami telah melakukan *survey* atau pengujian kepada 5 mahasiswa dengan memberikan sebanyak 5 pertanyaan mengenai *trainer kit control system and data acquisition* dengan tujuan untuk mengetahui apakah *trainer kit* ini membantu mahasiswa dalam pembelajaran *control system and data acquisition*. Lembar kuisisioner terlampir pada lampiran.

Berikut ini 5 pertanyaan yang diajukan serta persentase jawaban yang didapat.

Pertanyaan :

1. Seberapa penting *trainer kit control system and data acquisition* untuk digunakan dalam pembelajaran praktikum mahasiswa Polman Babel ?

Jawaban :

Tabel 4.5 Persentase Jawaban Kuisiner No.1

Penting	Cukup Penting	Tidak Penting
100 %	0	0

Pertanyaan :

2. Seberapa mudah alat ini digunakan ?

Jawaban :

Tabel 4.6 Persentase Jawaban Kuisiner No.2

Mudah	Cukup Mudah	Sulit
80 %	20 %	0

Pertanyaan :

3. Apakah alat ini praktis untuk pembelajaran *control system and data acquisition* ?

Jawaban :

Tabel 4.7 Persentase Jawaban Kuisiner No.3

Praktis	Cukup Praktis	Tidak Praktis
60 %	40 %	0

Pertanyaan :

4. Setelah menggunakan alat ini, seberapa paham anda mengenai pembelajaran *control system and data acquisition* ?

Jawaban :

Tabel 4.8 Persentase Jawaban Kuisiner No.4

Paham	Cukup Paham	Tidak Paham
80 %	20 %	0

Pertanyaan :

5. Apakah *trainer kit* ini layak memenuhi standar pembelajaran *control system and data acquisition* di Lab. Elektronika Polman Babel ?

Jawaban :

Tabel 4.9 Persentase Jawaban Kuisisioner No.5

Layak	Cukup Layak	Tidak Layak
80 %	20 %	0

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan beberapa tahapan yang sudah dilakukan sebelumnya mengenai proyek akhir ini. Maka diperoleh kesimpulan sebagai berikut :

1. Pengujian pada tiap-tiap komponen dilakukan satu persatu dan mendapatkan *output* berupa nilai dan grafik pada platform blynk.
2. Pada praktikum pengontrolan LED, pengontrolan nyala/mati pada keempat LED dapat dilakukan sesuai keinginan pengguna dan ditampilkan indikasinya pada platform blynk.
3. Pada praktikum pengontrolan *switch*, tampilan blynk hanya menampilkan kondisi saat praktikum *switch* dilakukan.
4. Pada praktikum pengontrolan suhu dan kelembaban, komponen sensor DHT22 memiliki tingkat kestabilan yang cukup tinggi.
5. Pada Sensor INA219, terdapat resistor yang berfungsi sebagai beban pada rangkaian serta komponen DC to DC *Converter* sebagai penaik dan penurun tegangan dan untuk mendapatkan nilai yang konstan yaitu 25V.
6. Pada Sensor Tegangan, digunakan komponen DC to DC *Converter* sebagai penaik dan penurun tegangan dan untuk mendapatkan nilai yang konstan yaitu 25V.
7. Pada praktikum pengontrolan *driver* motor & motor menggunakan 2 buah motor DC yang memiliki perbedaan fungsi. Untuk motor 1 digunakan untuk identifikasi putaran CW dan CCW, sedangkan motor 2 digunakan sebagai deteksi tegangan yang keluar selama motor berputar yang terhubung dengan sensor tegangan.

#### **5.2 Saran**

Berdasarkan hasil pengujian yang telah dilakukan, didapat beberapa saran sebagai pengembangan *trainer kit* kedepannya.

1. Menambahkan komponen lain pada *trainer kit* guna untuk memperluas cakupan pembelajaran *control system and data acquisition*.
2. Menambahkan fitur *history* melalui program agar dapat menyimpan data praktikum yang telah dilakukan sebelumnya.

## DAFTAR PUSTAKA

- [1] Cucuk, Agustinus, Diktat Sistem SCADA, Jakarta: PT PLN DISJAYA, 2014.
- [2] Muhammad Nurdin Abdul Muhaemin, David Setiadi, "PENERAPAN INTERNET OF THINGS (IoT) PADA SISTEM MONITORING IRIGASI (SMART IRIGASI)," *jurnal Infotronik*, vol. III, no. 2, p. 96, 2018.
- [3] F. Afrizal, Monitoring Model Sistem Pengepakan dan Penyortiran Barang Berbasis SCADA, Depok UI: D3 Fakultas Teknik Sarjana Elektronika, 2015.
- [4] w. kusbandono, "Pengertian Open Loop dan Close Loop," *speck of science*, 1 Maret 2013. [Online]. Available: <http://wisnukusbandono.blogspot.com/2013/03/pengertian-open-loop-dan-close-loop.html>. [Accessed 9 Agustus 2021].
- [5] Didik Aribowo, Mohammad Fatkhurrokhman, Hariyanto, "Pengembangan Media Pembelajaran Trainer Kit Pengendali Motor 3 Phase Pada Mata Pelajaran Instalasi Motor Listrik Di SMKN 4 Kota Serang," *Jurnal Pendidikan Teknik Elektro*, vol. V, no. 1, p. 3, 2020.
- [6] Muhammad Rifki Fauzi, Deni Ahmad Jakaria, "APLIKASI SMARTPHONE DENGAN PERINTAH SUARA UNTUK MENGENDALIKAN SAKLAR LISTRIK MENGGUNAKAN ARDUINO," *Jurnal Teknologi dan Informasi*, vol. VIII, no. 1, p. 23, 2020.
- [7] Ricky Merdeka, "Internet of Things (IoT) dengan ESP32 (Board D1- R32) untuk input Digital Sensor," Ricky Merdeka, [Online]. Available: <https://mrmerdeka.wordpress.com/2020/05/27/internet-of-things-iot-dengan-esp32-board-d1-r32-untuk-input-digital-sensor/>. [Accessed 27 May 2020].
- [8] Abdul Halim Mukti Nasution, Sri Indriani, Nida Fadhillah, Chandra Arifin, Saut Parsaoran Tamba, "PENGONTROLAN LAMPU JARAK JAUH DENGAN NODEMCU MENGGUNAKAN BLYNK," *jurnal TEKINKOM*, vol. II, no. 1, p. 94, 2019.

# **LAMPIRAN 1**

## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : Gazali Ma'rif  
Tempat Tanggal Lahir : Sungailiat, 5 Juni 2000  
Alamat : Gg. Singkur no 41,  
Lingkungan Nelayan 2  
Hp : 0852-6737-9675  
E-mail : gazalimarif@gmail.com  
Jenis Kelamin : Laki-laki  
Agama : Islam



### 2. Riwayat Pendidikan

Institusi	Program Studi	Tahun Lulus
SD 9 Sungailiat		2012
SMPN 1 Sungailiat		2015
SMKN 2 PangkalPinang	Teknik Elektronika Industri	2018

### 3. Pendidikan Non Formal : -

Sungailiat, 9 Agustus 2021

Penulis



## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : Winda Riani  
Tempat Tanggal Lahir : Bandung, 30 September  
2000  
Alamat : Jl. Kp.Jawa Atas,  
Sungailiat  
Hp : 085659670459  
E-mail : wriani300900@gmail.com  
Jenis Kelamin : Perempuan  
Agama : Islam



### 2. Riwayat Pendidikan

Institusi	Program Studi	Tahun Lulus
SDN Muhammadiyah		2012
SMPN 1 Sungailiat		2015
SMKN 1 Sungailiat	Multimedia	2018

### 3. Pendidikan Non Formal : -

Sungailiat, 9 Agustus 2021

Penulis

## **LAMPIRAN 2**

## **Program LED**

```
#include <Blynk.h>

#define BLYNK_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

WidgetLED led1(V26);

WidgetLED led2(V27);

WidgetLED led3(V28);

WidgetLED led4(V29);

char auth[] = "dJi22_5iTKPZTWJhGASeIxMcQ-ntnp0G";

char ssid[] = "cicakkabur";

char pass[] = "apasihkamu";

int LED1 = 18;

int LED2 = 19;

int LED3 = 23;

int LED4 = 5;

void setup()

{

  // Debug console

  Serial.begin(115200);

  pinMode(LED1, OUTPUT);

  pinMode(LED2, OUTPUT);

  pinMode(LED3, OUTPUT);
```

```
pinMode(LED4, OUTPUT);

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

BLYNK_WRITE(V21) {

  int pinValue = param.asInt();

  if (pinValue == 1) {

    digitalWrite(LED1, HIGH);

    led1.on();

  }

  else {

    digitalWrite(LED1, LOW); // Turn LED off

    led1.off();

  }

}

BLYNK_WRITE(V22) {

  int pinValue = param.asInt();

  if (pinValue == 1) {

    digitalWrite(LED2, HIGH);

    led2.on();

  }

  else {

    digitalWrite(LED2, LOW); // Turn LED off

    led2.off();

  }

}
```

```
BLYNK_WRITE(V23) {  
  int pinValue = param.asInt();  
  if (pinValue == 1) {  
    digitalWrite(LED3, HIGH);  
    led3.on();  
  }  
  else {  
    digitalWrite(LED3, LOW); // Turn LED off  
    led3.off();  
  }  
}  
  
BLYNK_WRITE(V24) {  
  int pinValue = param.asInt();  
  if (pinValue == 1) {  
    digitalWrite(LED4, HIGH);  
    led4.on();  
  }  
  else {  
    digitalWrite(LED4, LOW); // Turn LED off  
    led4.off();  
  }  
}  
  
void loop()  
{  
  Blynk.run();  
}
```

```
}
```

### **Program Switch**

```
#include <Blynk.h>

#define BLYNK_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

WidgetLCD lcd (V11);

const int pinButton = 25;

char auth[] = "dJi22_5iTKPZTWJhGASeIxMcQ-ntnp0G";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "cicakkabur";

char pass[] = "apasihkamu";

void setup() {

  Blynk.begin(auth, ssid, pass);

  pinMode(pinLED, OUTPUT);

  pinMode(pinButton, INPUT_PULLUP);

}

void loop() {

  Blynk.run();

  if(digitalRead(pinButton) == LOW){

    lcd.clear();

    lcd.print(0,0,"KONDISI SAAT INI");
```

```
lcd.print(0,1,">>SWITCH ON");  
  
  }else{  
  
    lcd.print(0,0,"KONDISI SAAT INI");  
  
    lcd.print(0,1,">>SWITCH OFF");  
  
  }  
  
}
```

### **Program Sensor Arus INA219**

```
#include <Blynk.h>  
  
#define BLYNK_PRINT Serial  
  
#include <WiFi.h>  
  
#include <WiFiClient.h>  
  
#include <BlynkSimpleEsp32.h>  
  
#include <Wire.h>  
  
#include <Adafruit_INA219.h>  
  
Adafruit_INA219 ina219;  
  
char auth[] = "dJi22_5iTKPZTWJhGASeIxMcQ-ntnp0G";  
  
char ssid[] = "cicakkabur";  
  
char pass[] = "apasihkamu";  
  
int enable1Pin = 26;  
  
const int freq = 3000;  
  
const int pwmChannel = 0;  
  
const int resolution = 8;
```

```
void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  while (!Serial) {
    delay(1);
  }
  ledcSetup(pwmChannel, freq, resolution);
  ledcAttachPin(enable1Pin, pwmChannel);
  uint32_t currentFrequency;
  Serial.println("Hello!");
  ina219.begin();
  Serial.println("Measuring voltage and current with INA219 ...");
}
BLYNK_WRITE(V2)
{
  int pinValue = param.asInt();
  ledcWrite(pwmChannel, pinValue);
  Blynk.virtualWrite(V1, pinValue);
  Serial.println(pinValue);
}
void loop(void)
{
  Blynk.run();
  Blynk.virtualWrite(V7, busvoltage);
```



```
Blynk.virtualWrite(V10,current_mA);  
  
delay(2000);  
  
}
```

### **Program Sensor Suhu DHT22**

```
#include "DHT.h"  
  
#define DHTPIN 12  
  
#define DHTTYPE DHT22  
  
DHT dht(DHTPIN, DHTTYPE);  
  
#define BLYNK_PRINT Serial  
  
#include <WiFi.h>  
  
#include <WiFiClient.h>  
  
#include <BlynkSimpleEsp32.h>  
  
char auth[] = "Te_GwKqUuhakioTauPrAVWeTSIoT09wA";  
  
char ssid[] = "cicakkabur";  
  
char pass[] = "apasihkamu";  
  
void setup() {  
  
  Blynk.begin(auth, ssid, pass);  
  
  Serial.begin(9600);  
  
  Serial.println("Deteksi Suhu dan Kelembaban");  
  
  Dht.begin();  
  
}  
  
void loop() {  
  
  delay(1000);  
  
  float t = dht.readSuhu();
```

```

float h = dht.readKelembaban();

if (isnan(h) || isnan(t)) {

  Serial.println("Semsor tidak terbaca!");

  return;

}

Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.println(" *C ");

Blynk.virtualWrite(V4,t);

Blynk.virtualWrite(V5,h);

}

```

### **Program Sensor Tegangan**

```

#include <Blynk.h>

#define BLYNK_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

//library ina 219

#include <Wire.h>

#include <Adafruit_INA219.h>

const int INA_addr = 0x40; // INA219 address

```

```
Adafruit_INA219 ina219(INA_addr);

float tegangan = 00;

float arus = 00;

float daya = 00;

char auth[] = "fk9oTwX5P8lHh7o4lFOSUd4lsT1I824D";

char ssid[] = "cicakkabur";

char pass[] = "apasihkamu";

const int voltage_sensor = 36;

float voltage_sensor_raw;

float v_beban;

int enable1Pin = 26;

const int freq = 3000;

const int pwmChannel = 0;

const int resolution = 8;

void setup()

{

  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);

  ledcSetup(pwmChannel, freq, resolution);

  ledcAttachPin(enable1Pin, pwmChannel);

  pinMode (voltage_sensor,INPUT);

}

BLYNK_WRITE(V2)

{

  int pinValue = param.asInt();
```

```

    ledcWrite(pwmChannel, pinValue);

    Blynk.virtualWrite(V1, pinValue);

    Serial.println(pinValue);
}

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

    baca_sensor();

    Serial.print ("NILAI TEGANGAN:");

    Serial.println (v_beban);

    Blynk.virtualWrite(V7,v_beban);

    delay(2000);
}

void baca_sensor()
{
    voltage_sensor_raw = analogRead(voltage_sensor);

    v_beban = map(voltage_sensor_raw,0,1023,0,2500);

    v_beban = v_beban/100*0.175;
}

```

### **Program *Driver*Motor & Motor**

```

#include <Blynk.h>

#define BLYNK_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

```

```
#include <BlynkSimpleEsp32.h>

char auth[] = "dJi22_5iTKPZTWJhGASeIxMcQ-ntnp0G";

char ssid[] = "cicakkabur";

char pass[] = "apasihkamu";

// sensor tegangan

const int voltage_sensor = 36;

float voltage_sensor_raw;

float v_beban;

int MOTORA = 17;

int MOTORB = 25;

int enable1Pin = 26;

// Setting PWM properties

const int freq = 30000;

const int pwmChannel = 0;

const int resolution = 8;

int dutyCycle = 170;

WidgetLCD lcd(V1);

void setup()

{

    //sensor tegangan

    pinMode (voltage_sensor,INPUT);

    Serial.begin(115200);

    pinMode(MOTORA, OUTPUT);

    pinMode(MOTORB, OUTPUT);

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

```

    ledcSetup(pwmChannel, freq, resolution);
    ledcAttachPin(enable1Pin, pwmChannel);
}
BLYNK_WRITE(V2)
{
    int pinValue = param.asInt();
    ledcWrite(pwmChannel, pinValue);
    Blynk.virtualWrite(V1, pinValue);
    Serial.println(pinValue);
}
BLYNK_WRITE(V3) {
    int pinValue = param.asInt();
    if (pinValue == 1) {
        digitalWrite(MOTORA, HIGH);
        digitalWrite(MOTORB, LOW);
        ledcWrite(pwmChannel, dutyCycle);
        dutyCycle = dutyCycle;
        lcd.clear();
        lcd.print(0,0,"Berputar CCW");
        Serial.print("Forward with duty cycle: ");
        Serial.println(dutyCycle);
    } else
    {
        digitalWrite(MOTORA, LOW);
        digitalWrite(MOTORB, LOW);
    }
}

```

```

    lcd.clear();

    lcd.print(0,0,"Motor Stop");
}
}

BLYNK_WRITE(V0) {
int pinValue = param.asInt();
if (pinValue == 1) {
    digitalWrite(MOTORA, LOW);
    digitalWrite(MOTORB, HIGH);
    ledcWrite(pwmChannel, dutyCycle);
    dutyCycle = dutyCycle;
    lcd.clear();
    lcd.print(0,0,"Berputar CW");
    Serial.print("Forward with duty cycle: ");
    Serial.println(dutyCycle);
} else
{
    digitalWrite(MOTORA, LOW);
    digitalWrite(MOTORB, LOW);
    lcd.clear();
    lcd.print(0,0,"Motor Stop");
}
}

void loop()
{

```

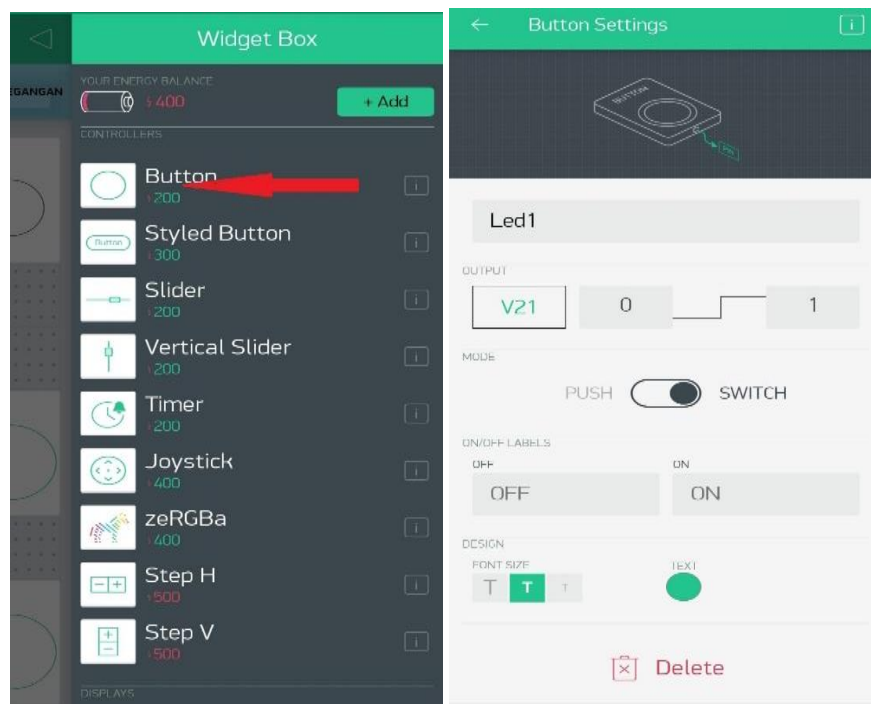
```
Blynk.run();  
  
// baca sensor tegangan  
  
baca_sensor();  
  
Serial.print ("NILAI TEGANGAN:");  
  
Serial.println (v_beban);  
  
Blynk.virtualWrite(V6,v_beban);  
  
delay(3000);  
  
}  
  
void baca_sensor()  
  
{  
  
voltage_sensor_raw = analogRead(voltage_sensor);  
  
v_beban = map(voltage_sensor_raw,0,1023,0,2500);  
  
v_beban = v_beban/100/5.6;  
  
}
```



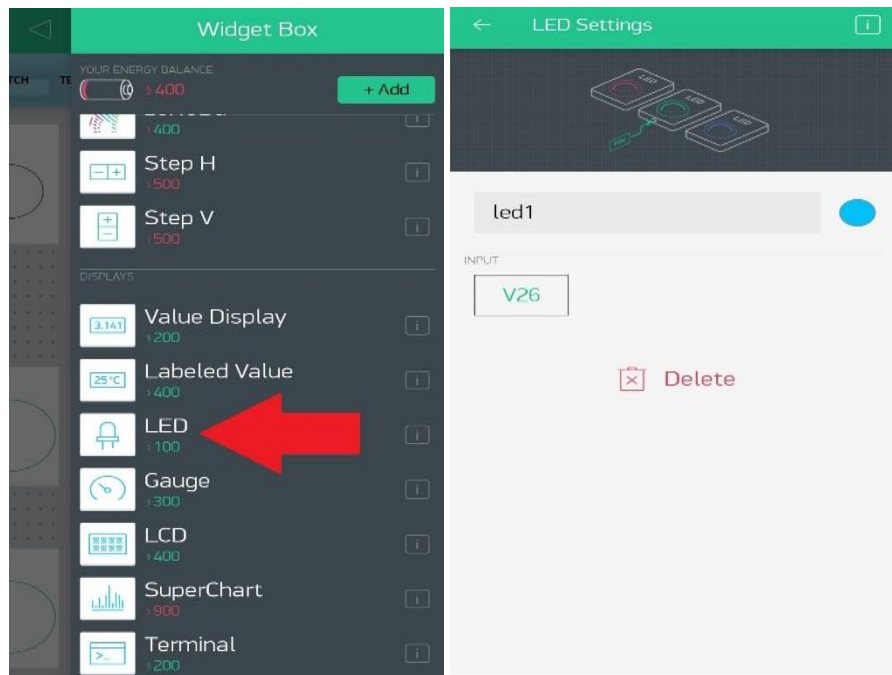
## **LAMPIRAN 3**

## Project 1- Pengontrolan LED

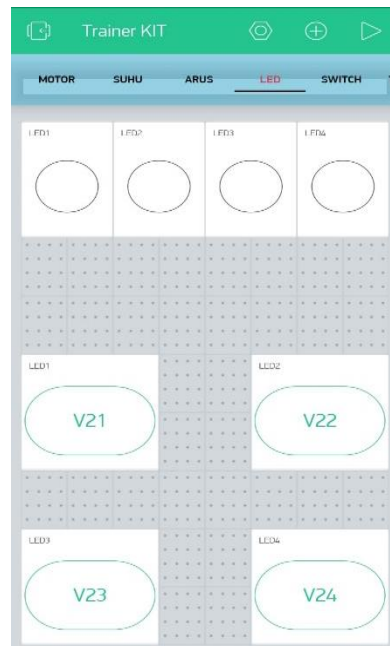
1. Buka menu *PlayStore* pada *smartphone*. Cari aplikasi BLYNK kemudian instal hingga selesai.
2. Buka aplikasi Blynk, maka akan muncul tampilan *Log In*. Klik *Create New Account*. Isikan alamat email dan isi pasword. Setelah di *Create*, maka aplikasi Blynk akan mengirimkan *authentication* token ke *email* yang dituliskan tadi.
3. Setelah *login*, pilih menu *New Project*. Pada tampilan *Create New Project* isikan nama *project* yang akan dibuat, pada *Choose Device* pilihlah perangkat yang digunakan contoh disini kita menggunakan esp32, dan pilih tipe koneksi yang akan digunakan contoh menggunakan *WiFi*. Untuk tema bisa memilih yang gelap atau terang.
4. Kemudian akan masuk ke papan *project*. Untuk menambahkan *widget*, pilih tombol *Add* atau tekan papan *project*.
5. Tekan *widget Button* maka akan masuk ke *Button Setting*. Tekan PIN untuk memilih pin ESP32. Ambil 4 button untuk mengontrol 4 buah LED, beri nama LED1-LED4. Lalu ubah pengaturannya seperti gambar dibawah. Sebagai contoh disini menggunakan PIN V21,V22,V23,dan V24.



6. Kemudian tambahkan lagi 4 buah *widget* LED sebagai indikator lalu ubah warna sesuai yang anda gunakan, pilih pin yang berbeda pada setiap *widget* lalu beri nama. Disini menggunakan PIN V26,V27,V28,dan V29 sebagai contoh :



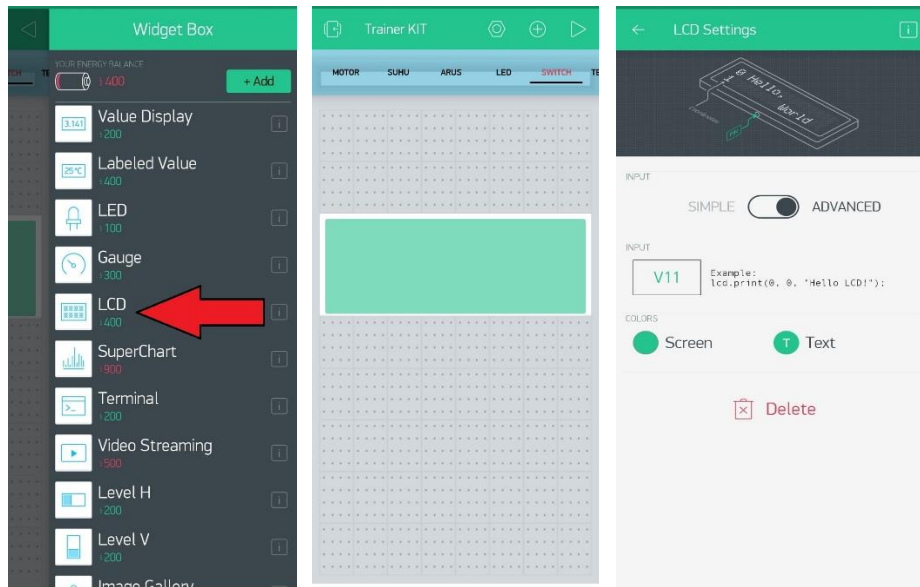
7. Tampilan Akhir akan menjadi seperti ini. Posisi dan bentuk *button* juga dapat anda atur.



8. Kemudian buka Aplikasi Arduino IDE, *download library* blynk dan esp32 bila belum punya. Kemudian klik “NEW” lalu masukkan program. Kemudian *copy authentication* token pada *email* kemudian *paste* pada Arduino IDE. Masukkan juga nama jaringan dan password yang digunakan agar ESP32 dapat masuk ke jaringan kita.
9. Kemudian hubungkan komponen LED pada Trainer Kit sesuai pin yang digunakan pada program.
10. Kemudian upload program hingga berhasil lalu buka aplikasi Blynk kemudian coba tekan button yang telah dibuat.

## Project 2– Pengontrolan Switch

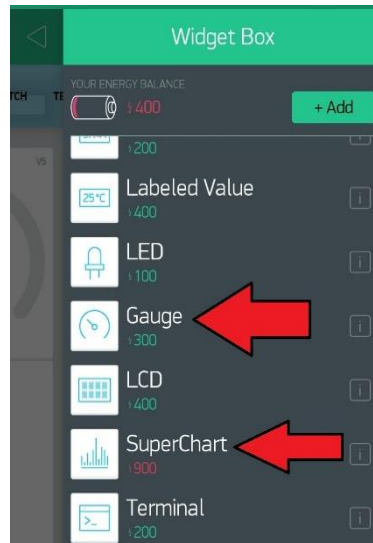
1. Pada Blynk pilih menu New Project lalu tulis nama *project*.
2. Kemudian pilih widget LCD, lalu ubah settingan ke *Advance* lalu pilih pin yang digunakan.



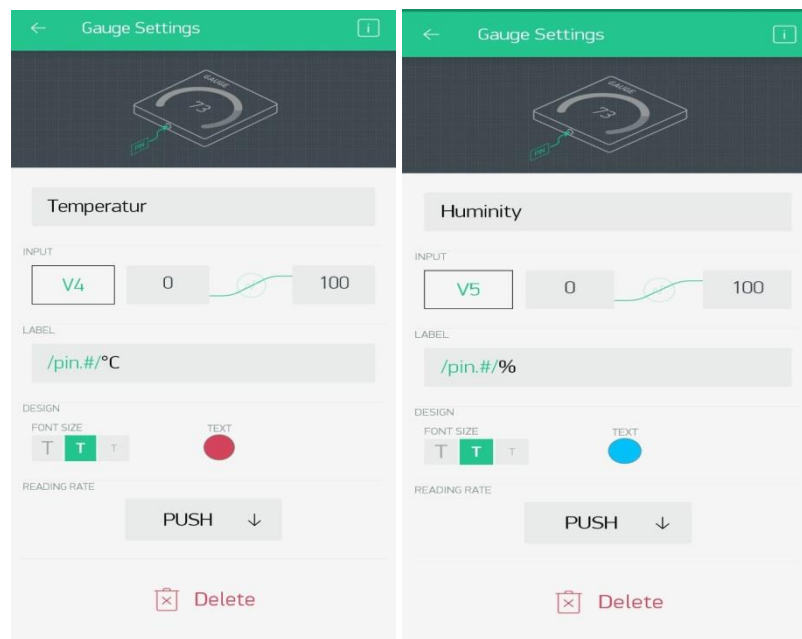
3. Lalu buka aplikasi Arduino IDE kemudian masukkan program.
4. Terakhir buka Aplikasi Blynk dan lihat tampilan LCD saat switch dimainkan pada trainer kit.

### Project 3 - Pengukuran Suhu dan Kelembaban

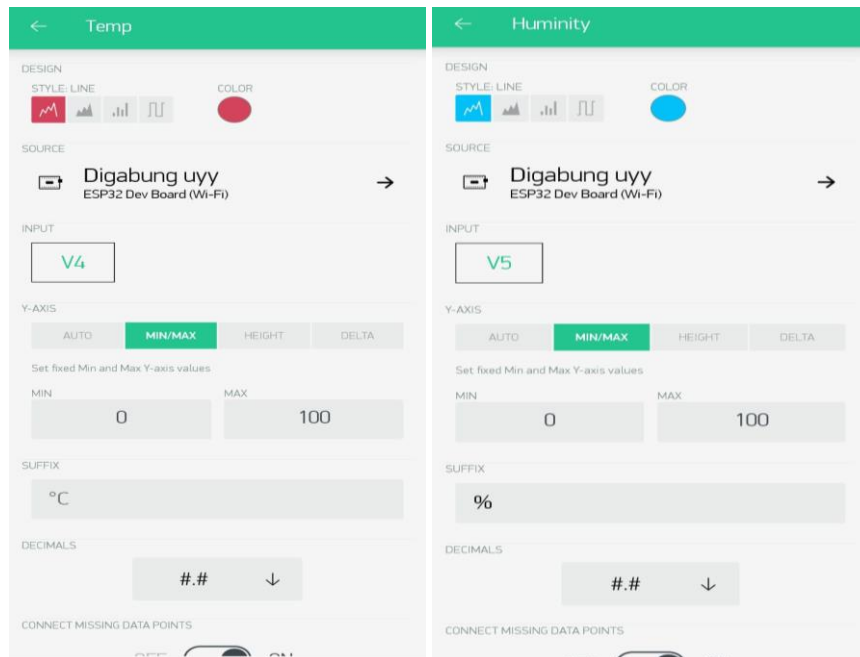
1. Pada Blynk pilih menu New Project lalu tulis nama *project*
2. Kemudian klik *Add* tambahkan *Widget Gauge* dan *Widget Super Chart* masing-masing 2. Ini digunakan sebagai monitoring nilai suhu dan kelembaban.



3. Kemudian *setting Gauge* 1 dan 2 seperti gambar di bawah ini. Disini *Gauge* 1 untuk tampilan *Temperatur* dan 2 untuk *Humidity*.



4. Kemudian setting kedua *Super Chart* seperti gambar dibawah ini, pilih pin yang sama pada *gauge* diatas, *widget* ini digunakan untuk menampilkan grafik nilai *Temperatur* dan *Humidity*.



5. Tampilan akhir akan menjadi seperti ini

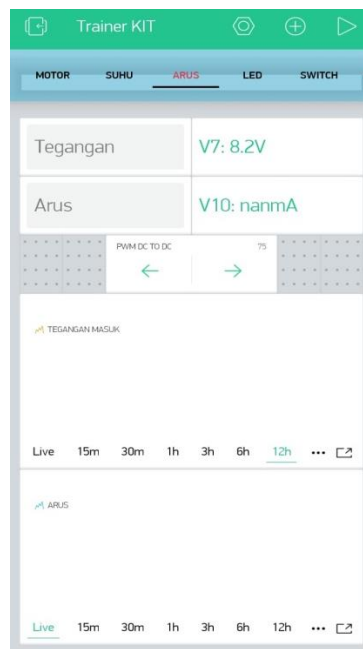


6. Kemudian Buka Aplikasi Arduino IDE lalu masukkan program.
7. Kemudian hubungkan modul sensor ke pin esp32 pada Trainer kit sesuai pin program yang anda gunakan. Kemudian upload program hingga berhasil lalu buka aplikasi Blynk kemudian coba lihat tampilan nilai dan grafik sensor



#### Project 4– Pengukuran Arus dan Tegangan dengan DC to Dc Converter

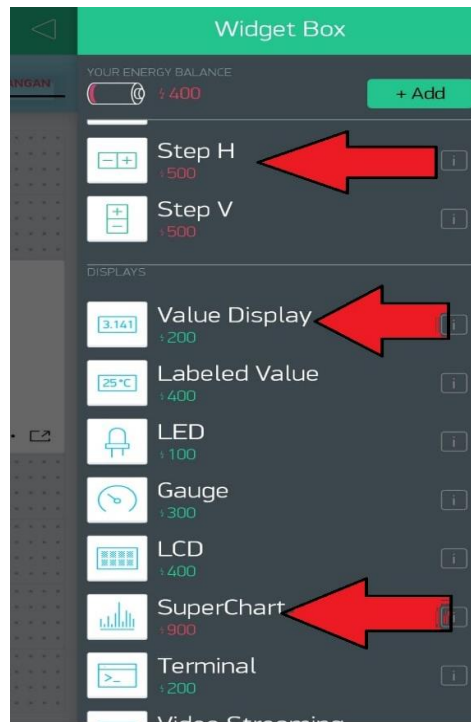
1. Pada Blynk pilih menu New Project lalu tulis nama project.
2. Lalu tambahkan *Widget Step H*, *Value Display* 2 buah, dan *Superchart* 2 buah lalu atur posisi sesuai kemauan anda.
3. Kemudian beri nama dan atur sama seperti praktikum sebelumnya sehingga menampilkan hasil akhir seperti gambar berikut. Terdapat 2 *widget* untuk menampilkan Tegangan dan Arus dan grafiknya juga dan juga *widget* untuk menaikkan tegangan pada DC to DC Converter.



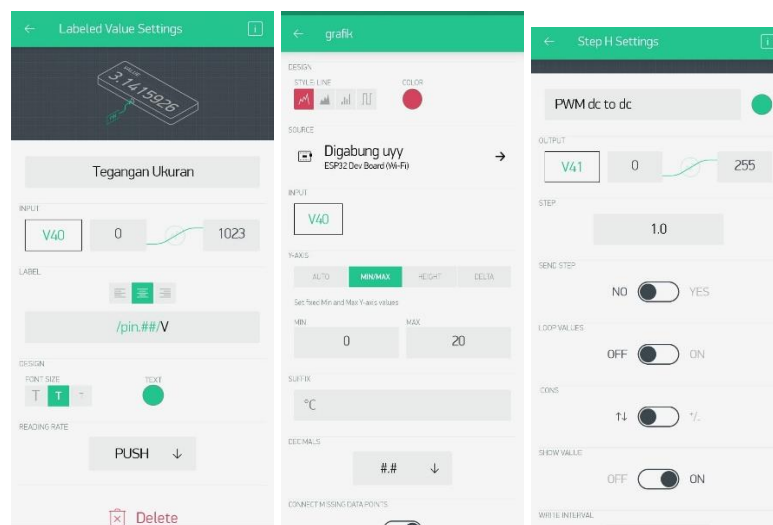
4. Kemudian buka Aplikasi Arduino IDE lalu masukkan program di bawah ini. Kemudian upload program hingga berhasil lalu buka aplikasi Blynk untuk melihat hasil pengukuran sensor dan untuk mengontrol naik turun tegangannya.

## Project 5 -Pengukuran Tegangan dengan DC to DC Converter

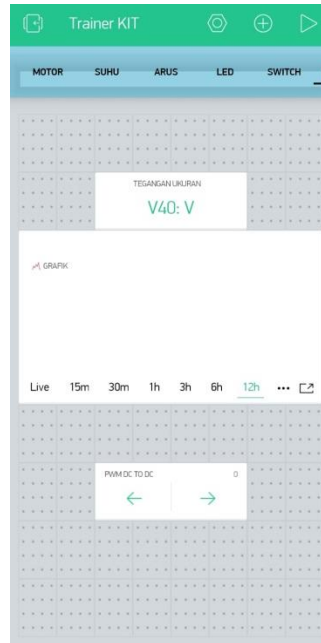
1. Pada Blynk pilih menu New Project tulis nama *project*.
2. Lalu tambahkan *Widget Step H*, *Value Display*, dan *Superchart* lalu atur posisi sesuai kemauan anda.



3. Kemudian ubah settingan masing masing Widget seperti gambar di bawah ini. Beri nama dan pilih pin yang digunakan.



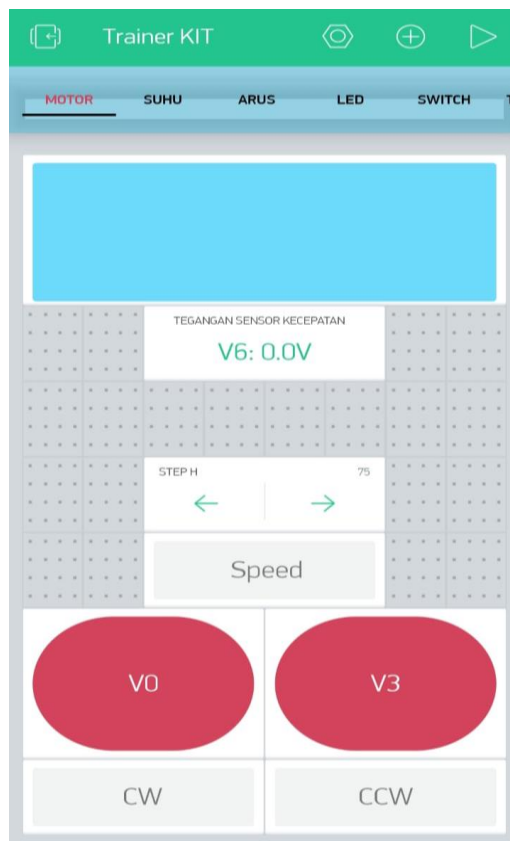
4. Tampilan keseluruhan seperti di bawah ini. Disini akan menampilkan nilai hasil ukuran sensor dan grafiknya, kemudian tegangan pada DC to DC Converter dapat kita naikan menggunakan *Widget Step H*.



5. Lalu rangkailah pada Trainer Kit.
6. Kemudian Buka Aplikasi Arduino IDE lalu masukkan program.
7. Kemudian upload program hingga berhasil lalu buka aplikasi Blynk untuk melihat hasil pengukuran sensor dan untuk mengontrol naik turun tegangannya.

## Project 6-Pengontrolan Motor

1. Pada Blynk pilih menu New Project lalu tulis nama *project*.
2. Kemudian Masukkan *Widget LCD* (untuk tampilan putaran motor), *Value Display* (untuk sensor tegangan motor), *Step H*( Untuk menaikkan atau menurunkan kecepatan motor), dan 2 buah *Button* untuk kontrol arah putar motor. Buat seperti gambar dibawah kemudian pilih PIN yang ingin and gunakan pada setiap *Widget*.



3. Kemudian Buka Aplikasi Arduino IDE dan masukkan program berikut untuk mengontrol dan memonitoring motor kemudian upload ke trainer kit.
4. Terakhir buka Aplikasi Blynk dan jalankan. Disitu anda dapat mengontrol dan memonitoring motor.

## **LAMPIRAN 4**

KUISONER *TRAINER KIT CONTROL AND DATA ACQUISITION*

Nama Responden : PUTRI ANDREANI


Kelas : 3 EA

Petunjuk pengisian

Berilah tanda [✓] pada kolom yang sudah disediakan.

1. Seberapa penting *trainer kit control system and data acquisition* untuk digunakan dalam pembelajaran praktikum mahasiswa Polman Babel ?  
[✓] Penting            [ ] Cukup Penting            [ ] Tidak Penting
2. Seberapa mudah alat ini digunakan ?  
[✓] Mudah            [ ] Cukup Mudah            [ ] Sulit
3. Apakah alat ini praktis untuk pembelajaran *control system and data acquisition* ?  
[X] Praktis            [✓] Cukup Praktis            [ ] Tidak Praktis
4. Setelah menggunakan alat ini, seberapa paham anda mengenai pembelajaran *control system and data acquisition* ?  
[✓] Paham            [ ] Cukup Paham            [ ] Tidak Paham
5. Apakah *trainer kit* ini layak memenuhi standar pembelajaran *control system and data acquisition* di Lab. Elektronika Polman Babel ?  
[✓] Layak            [ ] Cukup Layak            [ ] Tidak Layak

Responden

  
Putri Andreani

KUISONER *TRAINER KIT CONTROL AND DATA ACQUISITION*

Nama Responden : NILA FAZILA

Kelas : 3EA

Petunjuk pengisian

Berilah tanda [✓] pada kolom yang sudah disediakan.

1. Seberapa penting *trainer kit control system and data acquisition* untuk digunakan dalam pembelajaran praktikum mahasiswa Polman Babel ?  
[✓] Penting            [ ] Cukup Penting            [ ] Tidak Penting
2. Seberapa mudah alat ini digunakan ?  
[✓] Mudah            [ ] Cukup Mudah            [ ] Sulit
3. Apakah alat ini praktis untuk pembelajaran *control system and data acquisition* ?  
[✓] Praktis            [ ] Cukup Praktis            [ ] Tidak Praktis
4. Setelah menggunakan alat ini, seberapa paham anda mengenai pembelajaran *control system and data acquisition* ?  
[✓] Paham            [ ] Cukup Paham            [ ] Tidak Paham
5. Apakah *trainer kit* ini layak memenuhi standar pembelajaran *control system and data acquisition* di Lab. Elektronika Polman Babel ?  
[✓] Layak            [ ] Cukup Layak            [ ] Tidak Layak

Responden

  
NILA F.

KUISONER *TRAINER KIT CONTROL AND DATA ACQUISITION*

Nama Responden : Ninda Puspita

Kelas : 2 EB

Petunjuk pengisian

Berilah tanda [✓] pada kolom yang sudah disediakan.

1. Seberapa penting *trainer kit control system and data acquisition* untuk digunakan dalam pembelajaran praktikum mahasiswa Polman Babel ?  
[✓] Penting            [ ] Cukup Penting            [ ] Tidak Penting
2. Seberapa mudah alat ini digunakan ?  
[✓] Mudah            [ ] Cukup Mudah            [ ] Sulit
3. Apakah alat ini praktis untuk pembelajaran *control system and data acquisition* ?  
[✓] Praktis            [ ] Cukup Praktis            [ ] Tidak Praktis
4. Setelah menggunakan alat ini, seberapa paham anda mengenai pembelajaran *control system and data acquisition* ?  
[✓] Paham            [ ] Cukup Paham            [ ] Tidak Paham
5. Apakah *trainer kit* ini layak memenuhi standar pembelajaran *control system and data acquisition* di Lab. Elektronika Polman Babel ?  
[✓] Layak            [ ] Cukup Layak            [ ] Tidak Layak

Responden



Ninda Puspita.



KUISONER *TRAINER KIT CONTROL AND DATA ACQUISITION*

Nama Responden : Rafii Pratama


Kelas : 2 E 13

Petunjuk pengisian

Berilah tanda [✓] pada kolom yang sudah disediakan.

1. Seberapa penting *trainer kit control system and data acquisition* untuk digunakan dalam pembelajaran praktikum mahasiswa Polman Babel ?  
[✓] Penting            [ ] Cukup Penting            [ ] Tidak Penting
2. Seberapa mudah alat ini digunakan ?  
[✓] Mudah            [ ] Cukup Mudah            [ ] Sulit
3. Apakah alat ini praktis untuk pembelajaran *control system and data acquisition* ?  
[✓] Praktis            [ ] Cukup Praktis            [ ] Tidak Praktis
4. Setelah menggunakan alat ini, seberapa paham anda mengenai pembelajaran *control system and data acquisition* ?  
[ ] Paham            [✓] Cukup Paham            [ ] Tidak Paham
5. Apakah *trainer kit* ini layak memenuhi standar pembelajaran *control system and data acquisition* di Lab. Elektronika Polman Babel ?  
[✓] Layak            [ ] Cukup Layak            [ ] Tidak Layak

Responden

  
Rafii Pratama

KUISONER *TRAINER KIT CONTROL AND DATA ACQUISITION*

Nama Responden : SALSABILA MAHARENI

Kelas : 3EA

Petunjuk pengisian

Berilah tanda [√] pada kolom yang sudah disediakan.

1. Seberapa penting *trainer kit control system and data acquisition* untuk digunakan dalam pembelajaran praktikum mahasiswa Polman Babel ?  
 Penting             Cukup Penting             Tidak Penting
2. Seberapa mudah alat ini digunakan ?  
 Mudah             Cukup Mudah             Sulit
3. Apakah alat ini praktis untuk pembelajaran *control system and data acquisition* ?  
 Praktis             Cukup Praktis             Tidak Praktis
4. Setelah menggunakan alat ini, seberapa paham anda mengenai pembelajaran *control system and data acquisition* ?  
 Paham             Cukup Paham             Tidak Paham
5. Apakah *trainer kit* ini layak memenuhi standar pembelajaran *control system and data acquisition* di Lab. Elektronika Polman Babel ?  
 Layak             Cukup Layak             Tidak Layak

Responden

  
Salsabila Mahareni

# **LAMPIRAN 5**

MODUL PRAKTIKUM *SYSTEM CONTROL AND  
DATA ACQUISITION*

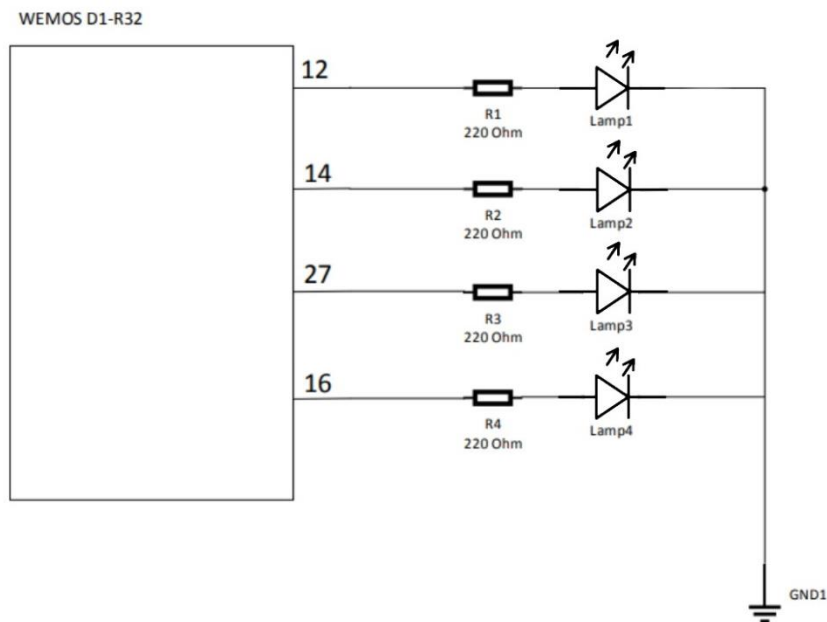


POLITEKNIK MANUFAKTUR NEGERI BANGKA  
BELITUNG

# PERCOBAAN 1

## KONTROL LED

### 1. Gambar Rangkaian

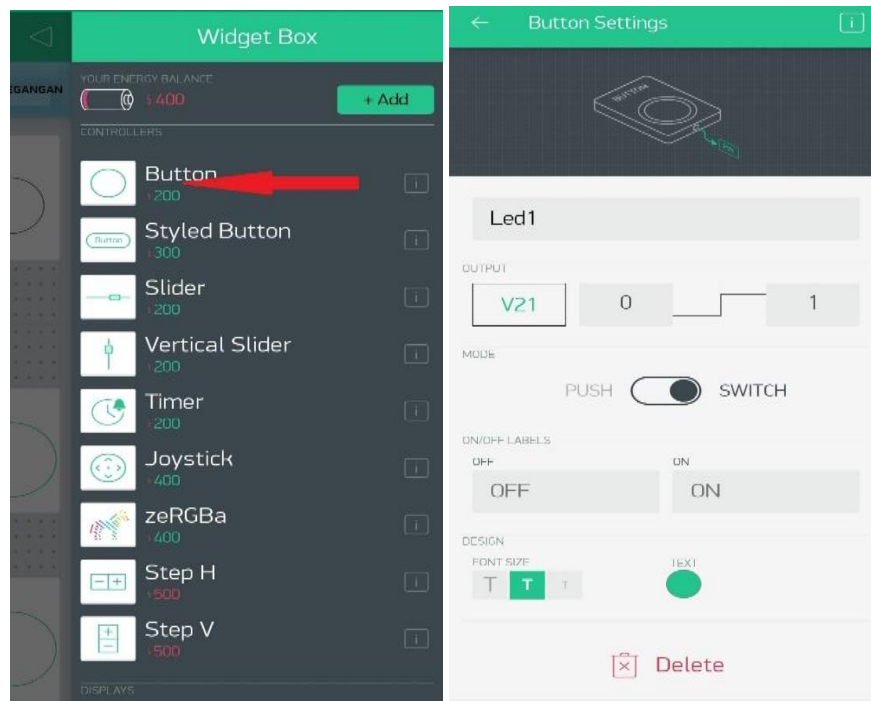


### 2. Prosedur Percobaan

- Sambungkan tiap-tiap kaki led ke pin serta *ground* seperti gambar rangkaian diatas menggunakan kabel jumper.
- Buat program untuk menghidupkan LED pada Arduino IDE.
- Untuk menampilkan pada blynk, silahkan membuat program untuk tampilan blynk pada Arduino IDE. Jangan lupa juga untuk membuat *widget* pada platform blynk. Ikuti langkah dibawah ini.
  - Buka menu *PlayStore* pada *smartphone*. Cari aplikasi BLYNK kemudian instal hingga selesai.
  - Buka aplikasi Blynk, maka akan muncul tampilan *Log In*. Klik *Crate New Account*. Isikan alamat email dan isi pasword. Setelah

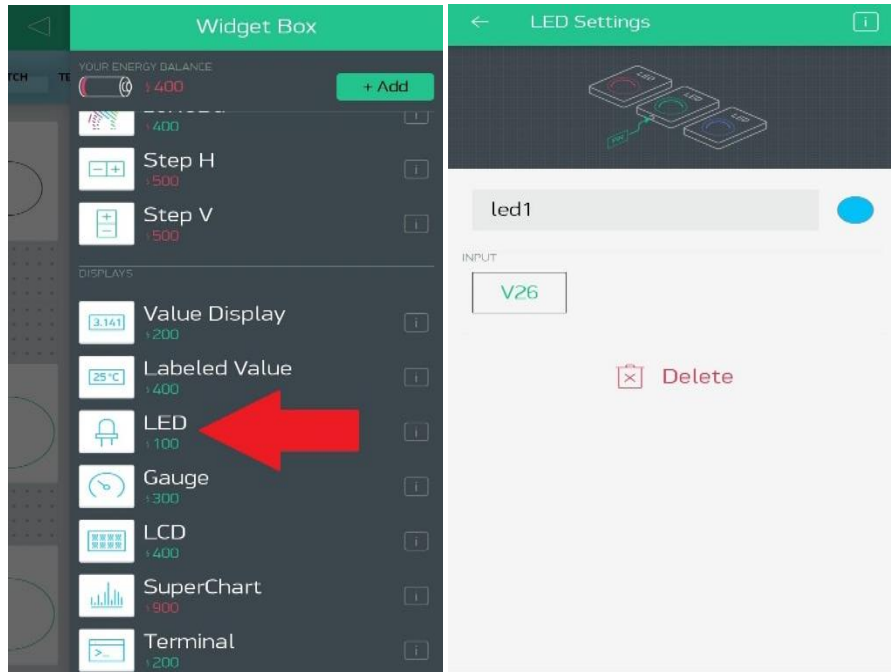
di *Create*, maka aplikasi Blynk akan mengirimkan *authentication* token ke *email* yang dituliskan tadi.

- Setelah *login*, pilih menu *New Project*. Pada tampilan *Create New Project* isikan nama *project* yang akan dibuat, pada *Choose Device* pilihlah perangkat yang digunakan contoh disini kita menggunakan esp32, dan pilih tipe koneksi yang akan digunakan contoh menggunakan *WiFi*. Untuk tema bisa memilih yang gelap atau terang.
- Kemudian akan masuk ke papan *project*. Untuk menambahkan *widget*, pilih tombol *Add* atau tekan papan *project*.
- Tekan *widget Button* maka akan masuk ke *Button Setting*. Tekan PIN untuk memilih pin ESP32. Ambil 4 button untuk mengontrol 4 buah LED, beri nama LED1-LED4. Lalu ubah pengaturannya seperti gambar dibawah. Sebagai contoh disini menggunakan PIN V21,V22,V23,dan V24.

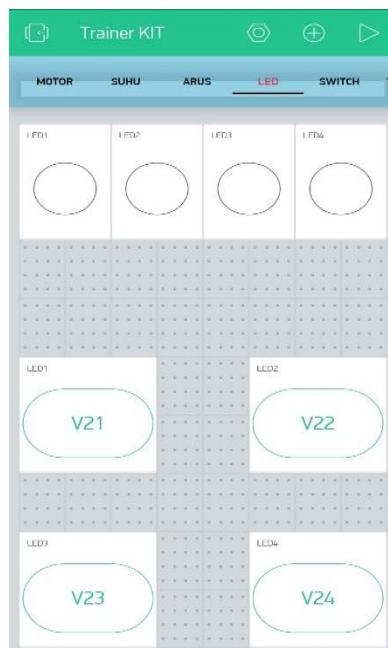


- Kemudian tambahkan lagi 4 buah *widget* LED sebagai indikator lalu ubah warna sesuai yang anda gunakan, pilih pin yang berbeda

pada setiap *widget* lalu beri nama. Disini menggunakan PIN V26,V27,V28,dan V29 sebagai contoh :



- Tampilan Akhir akan menjadi seperti ini. Posisi dan bentuk *button* juga dapat anda atur.



- Kemudian buka Aplikasi Arduino IDE, *download library* blynk dan esp32 bila belum punya. Kemudian klik “NEW” lalu masukkan program. Kemudian *copy authentication* token pada *email* kemudian *paste* pada Arduino IDE. Masukkan juga nama jaringan dan password yang digunakan agar ESP32 dapat masuk ke jaringan kita.
  - Kemudian hubungkan komponen LED pada Trainer Kit sesuai pin yang digunakan pada program.
  - Kemudian upload program hingga berhasil lalu buka aplikasi Blynk kemudian coba tekan button yang telah dibuat.
- d. Isilah data tabel dibawah ini untuk mengidentifikasi nyala dan mati pada LED.

### 3. Tabel Hasil Percobaan

Nomor LED	Keterangan	
	ON	OFF
LED 1		
LED 2		
LED 3		
LED 4		

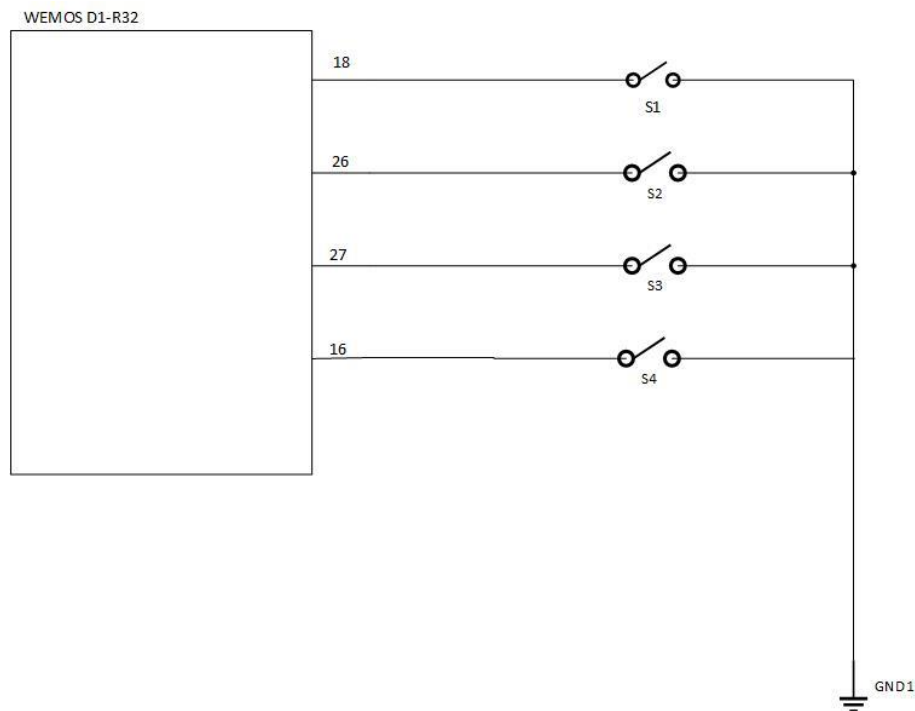
4. Buatlah analisa dari percobaan diatas.



## PERCOBAAN 2

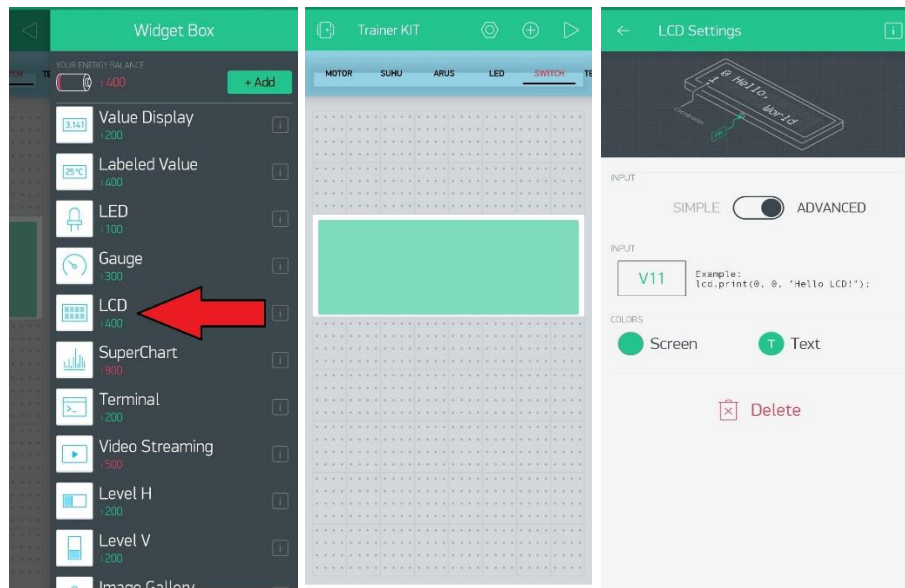
### KONTROL SWITCH

#### 1. Gambar Rangkaian



#### 2. Prosedur Percobaan

- a. Hubungkan switch ke pin *controller* serta ke GND.
- b. Buat program pada Arduino IDE
- c. Untuk menampilkan pada blynk, silahkan membuat program untuk tampilan blynk pada Arduino IDE. Jangan lupa juga untuk membuat widget pada platform blynk. Ikuti langkah dibawah ini.
  - Pada Blynk pilih menu New Project lalu tulis nama *project*.
  - Kemudian pilih widget LCD, lalu ubah settingan ke *Advance* lalu pilih pin yang digunakan.



- Lalu buka aplikasi Arduino IDE kemudian masukkan program.
- Terakhir buka Aplikasi Blynk dan lihat tampilan LCD saat switch dimainkan pada trainer kit.

d. Isilah data tabel dibawah ini

### 3. Tabel Hasil Percobaan

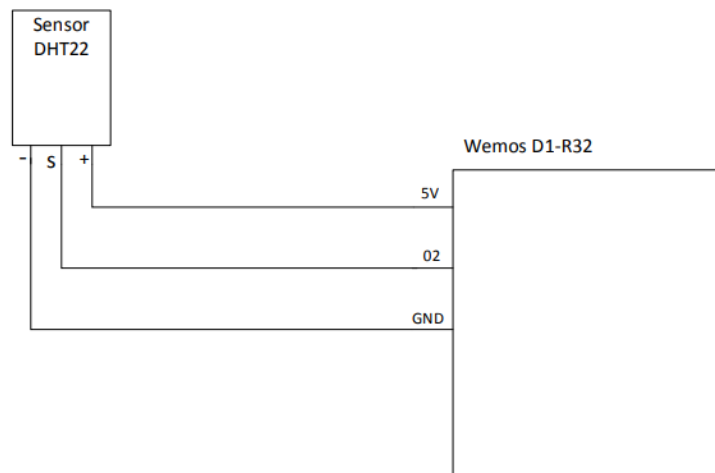
Switch	ON	OFF
1		
2		
3		
4		

4. Buatlah analisa dari percobaan diatas.

# PERCOBAAN 3

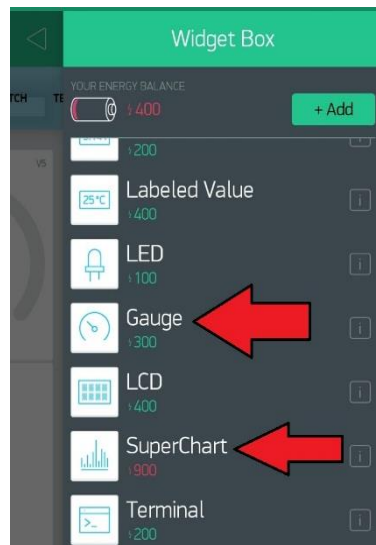
## PENGUKURAN SUHU DAN KELEMBABAN DENGAN SENSOR SUHU

### 1. Gambar Rangkaian

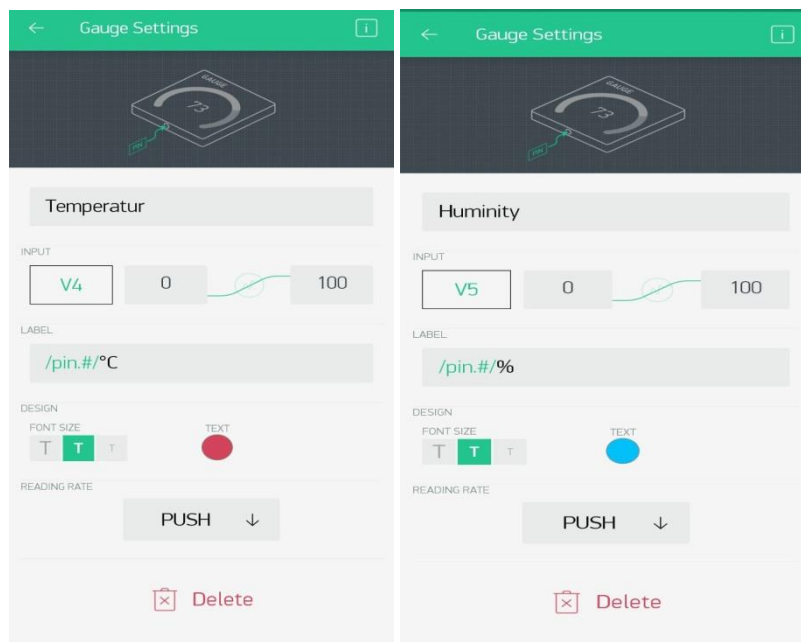


### 2. Prosedur Percobaan

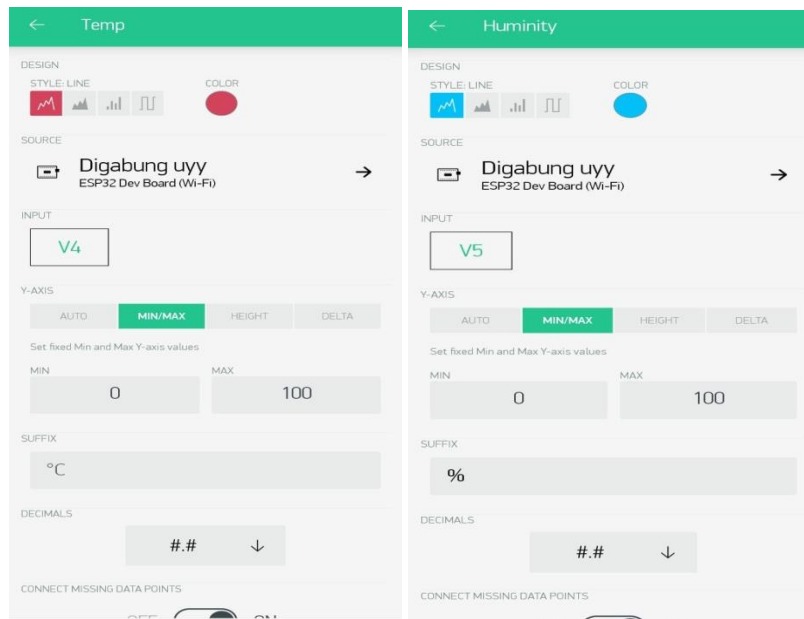
- a. Sambungkan tiap-tiap pin pada sensor suhu ke pin pada controller seperti gambar diatas.
- b. Buat program pada Arduino untuk mendapat data suhu yang keluar
- c. Untuk menampilkan pada blynk, silahkan membuat program untuk tampilan blynk pada Arduino IDE. Jangan lupa juga untuk membuat widget pada platform blynk. Ikuti langkah dibawah ini.
  - Pada Blynk pilih menu New Project lalu tulis nama *project*
  - Kemudian klik *Add* tambahkan *Widget Gauge* dan *Widget Super Chart* masing-masing 2. Ini digunakan sebagai monitoring nilai suhu dan kelembaban.



- Kemudian *setting Gauge* 1 dan 2 seperti gambar di bawah ini. Disini *Gauge* 1 untuk tampilan *Temperatur* dan 2 untuk *Humidity*.



- Kemudian *setting* kedua *Super Chart* seperti gambar dibawah ini, pilih pin yang sama pada *gauge* diatas, *widget* ini digunakan untuk menampilkan grafik nilai *Temperatur* dan *Humidity*.



- Tampilan akhir akan menjadi seperti ini



- Kemudian Buka Aplikasi Arduino IDE lalu masukkan program. Kemudian hubungkan modul sensor ke pin esp32 pada Trainer kit sesuai pin program yang anda gunakan. Kemudian upload program hingga berhasil lalu buka aplikasi Blynk kemudian coba lihat tampilan nilai dan grafik sensor.

- d. Ambil data setiap 5 detik sekali dalam waktu 1 menit dan isilah data pada tabel dibawah ini.

3. Tabel Hasil Percobaan

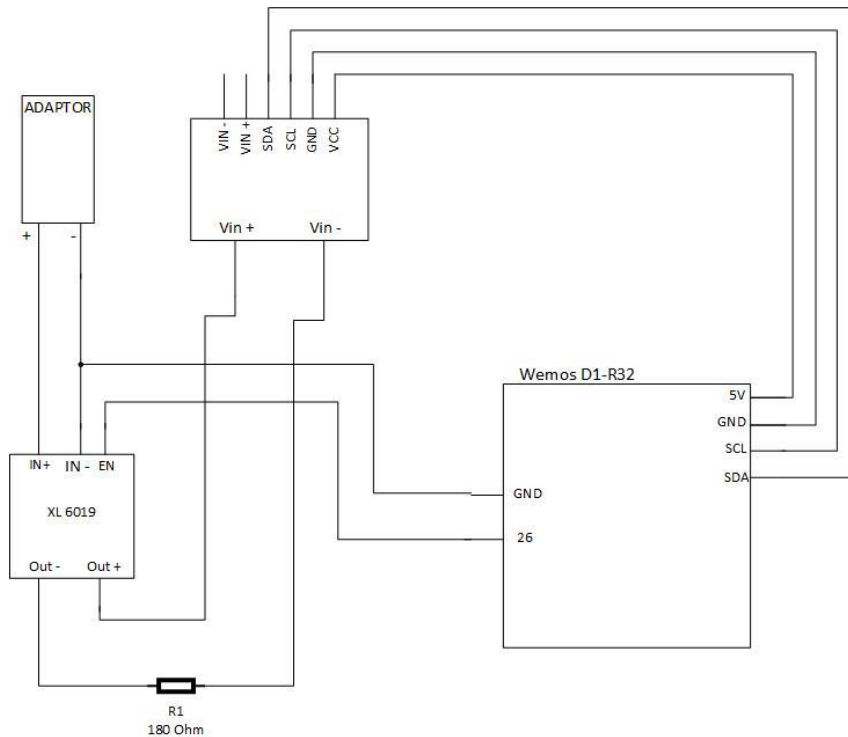
Waktu (s)	Temperatur	Humidity
5		
10		
15		
20		
25		
30		
35		
40		
45		
50		
55		
60		

4. Buatlah analisa dari percobaan diatas.

# PERCOBAAN 4

## PENGUKURAN TEGANGAN DAN ARUS DENGAN SENSOR INA219

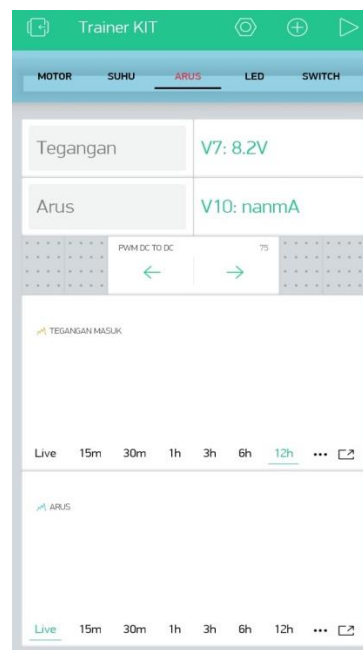
### 1. Gambar Rangkaian



### 2. Prosedur Percobaan

- Hubungkan setiap kaki-kaki ke pin seperti gambar diatas.
- Silahkan hubungkan adaptor yang belum tersambung ke stop kontak terlebih dahulu. Setelah sambungan benar adaptor dapat dihubungkan ke stop kontak.
- Buat program pada Arduino IDE.
- Untuk menampilkan pada blynk, silahkan membuat program untuk tampilan blynk pada Arduino IDE. Jangan lupa juga untuk membuat widget pada platform blynk. Ikuti langkah dibawah ini.
  - Pada Blynk pilih menu New Project lalu tulis nama project.


- Lalu tambahkan *Widget Step H*, *Value Display* 2 buah, dan *Superchart* 2 buah lalu atur posisi sesuai kemauan anda.
- Kemudian beri nama dan atur sama seperti praktikum sebelumnya sehingga menampilkan hasil akhir seperti gambar berikut. Terdapat 2 *widget* untuk menampilkan Tegangan dan Arus dan grafiknya juga dan juga *widget* untuk menaikkan tegangan pada DC to DC Converter.



- Kemudian buka Aplikasi Arduino IDE lalu masukkan program di bawah ini. Kemudian upload program hingga berhasil lalu buka aplikasi Blynk untuk melihat hasil pengukuran sensor dan untuk mengontrol naik turun tegangannya.
- e. Setelah selesai, ambil data pengukuran tegangan dan arus pada serial monitor dan multimeter.



3. Table Hasil Percobaan

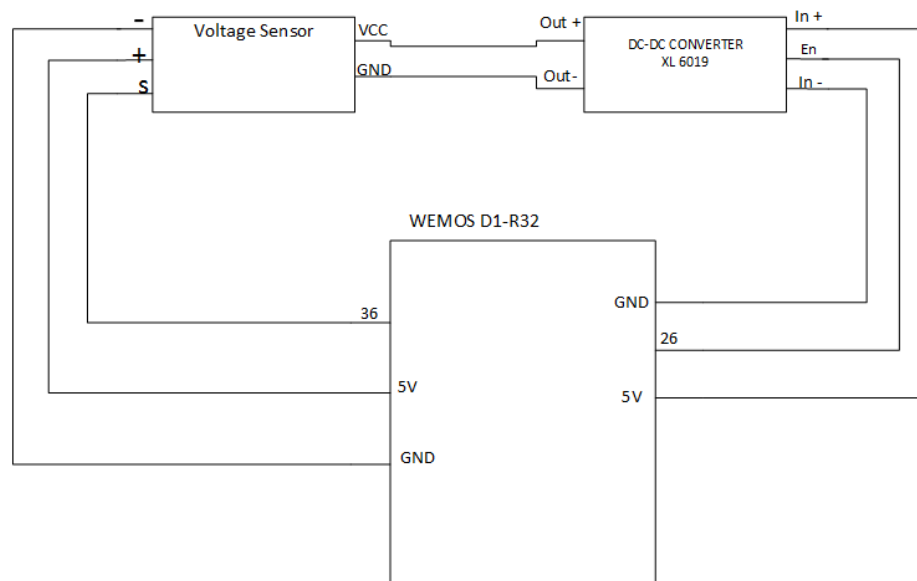
PWM	Nilai Tegangan (V)	Nilai Arus (mA)
0		
5		
10		
		
255		

4. Buatlah analisa dari percobaan diatas.

# PERCOBAAN 5

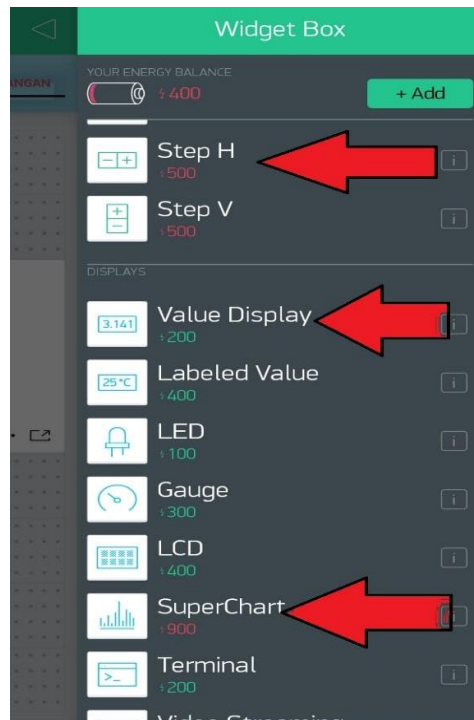
## PENGUKURAN TEGANGAN DENGAN SENSOR TEGANGAN

### 1. Gambar Rangkaian

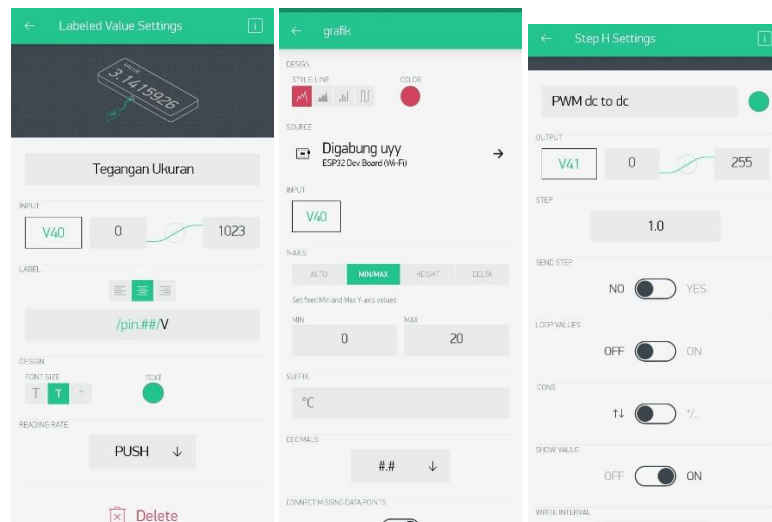


### 2. Prosedur Percobaan

- Hubungkan setiap kaki-kaki komponen ke pin wemos seperti gambar diatas.
- Buat program pada Arduino IDE.
- Untuk menampilkan pada blynk, silahkan membuat program untuk tampilan blynk pada Arduino IDE. Jangan lupa juga untuk membuat widget pada platform blynk. Ikuti langkah dibawah ini.
  - Pada Blynk pilih menu New Project tulis nama *project*.
  - Lalu tambahkan *Widget Step H*, *Value Display*, dan *Superchart* lalu atur posisi sesuai kemauan anda.

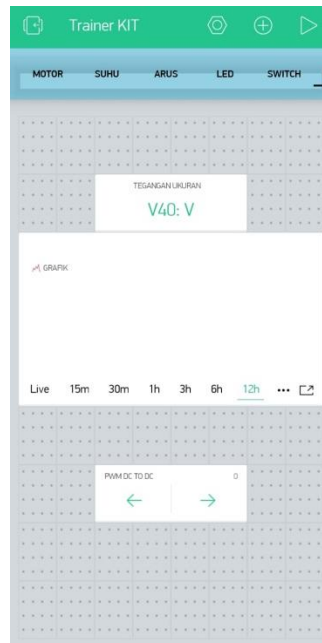


- Kemudian ubah settingan masing masing Widget seperti gambar di bawah ini. Beri nama dan pilih pin yang digunakan.




- Tampilan keseluruhan seperti di bawah ini. Disini akan menampilkan nilai hasil ukuran sensor dan grafiknya,

kemudian tegangan pada DC to DC *Converter* dapat kita naikan menggunakan *Widget Step H*.



- Lalu rangkailah pada Trainer Kit.
  - Kemudian Buka Aplikasi Arduino IDE lalu masukkan program.
  - Kemudian upload program hingga berhasil lalu buka aplikasi Blynk untuk melihat hasil pengukuran sensor dan untuk mengontrol naik turun tegangannya.
- d. Setelah selesai, ambil data pengukuran tegangan pada blynk dan multimeter dan isi tabel dibawah ini

3. Tabel Hasil Percobaan

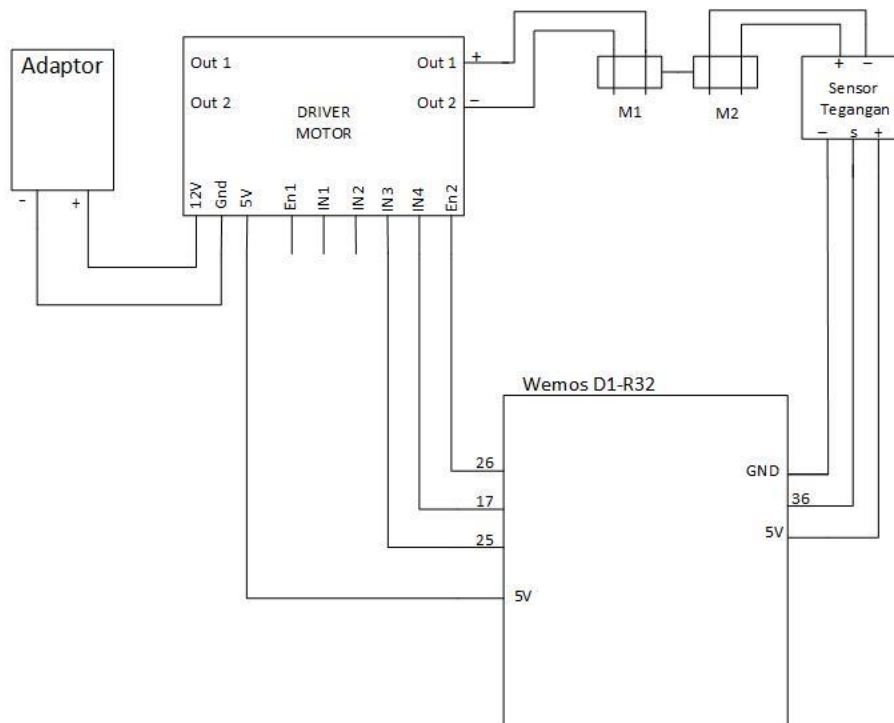
PWM	Nilai pada Blynk (V)	Nilai pada Multimeter (V)
0		
5		
10		
		
255		

4. Buatlah analisa dari percobaan diatas.

## PERCOBAAN 6

### DRIVER MOTOR & MOTOR

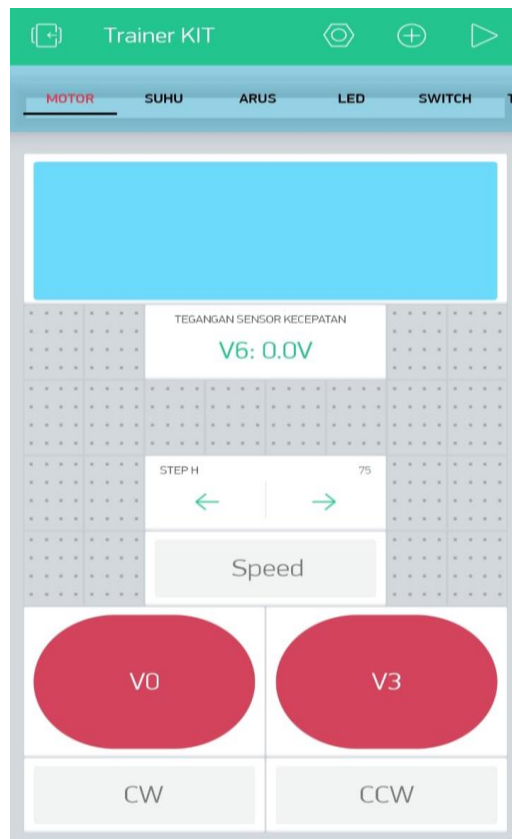
#### 1. Gambar Rangkaian



#### 2. Prosedur Percobaan


- Hubungkan kaki-kaki tiap komponen seperti gambar rangkaian diatas
- Silahkan hubungkan adaptor yang belum tersambung ke stop kontak terlebih dahulu. Setelah sambungan benar adaptor dapat dihubungkan ke stop kontak.
- Buat program pada Arduino IDE.
- Untuk menampilkan pada blynk, silahkan membuat program untuk tampilan blynk pada Arduino IDE. Jangan lupa juga untuk membuat widget pada platform blynk. Ikuti langkah dibawah ini.
  - Pada Blynk pilih menu New Project lalu tulis nama *project*.

- Kemudian Masukkan *Widget* LCD (untuk tampilan putaran motor), *Value Display* (untuk sensor tegangan motor), *Step H*( Untuk menaikkan atau menurunkan kecepatan motor), dan 2 buah *Button* untuk kontrol arah putar motor. Buat seperti gambar dibawah kemudian pilih PIN yang ingin and gunakan pada setiap *Widget*.



- Kemudian Buka Aplikasi Arduino IDE dan masukkan program berikut untuk mengontrol dan memonitoring motor kemudian upload ke trainer kit.  
Terakhir buka Aplikasi Blynk dan jalankan. Disitu anda dapat mengontrol dan memonitoring motor
- e. Setelah selesai, ambil data pengukuran tegangan pada blynk dan multimeter dan isi tabel dibawah ini.

3. Tabel Hasil Percobaan

PWM	Nilai Tegangan pada Blynk (V)	Nilai Tegangan pada Multimeter (V)	Error %
0			
5			
10			
			
255			

4. Buatlah analisa dari percobaan diatas.