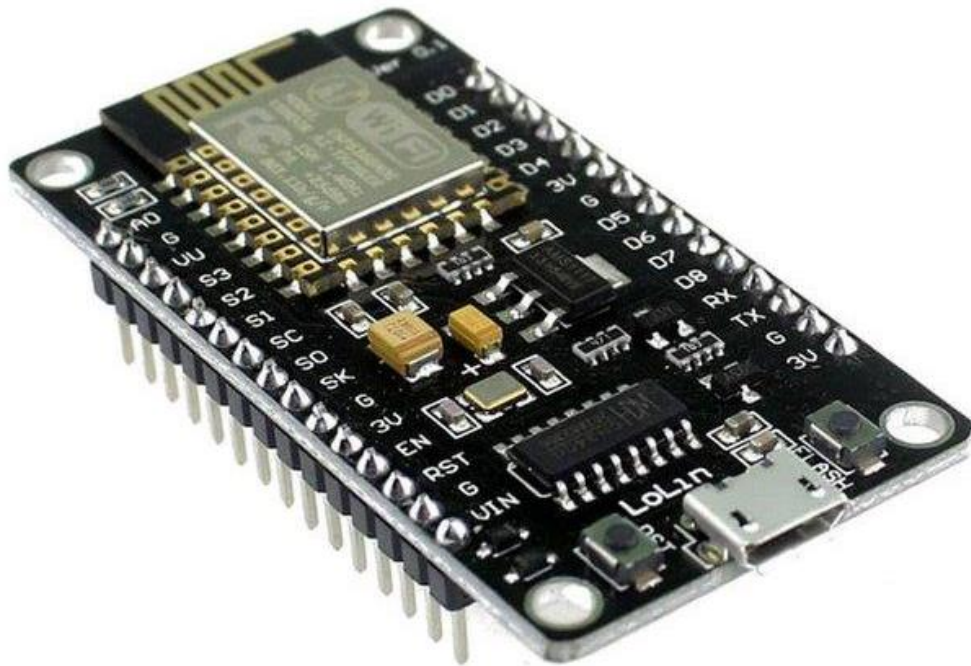


TRAINING MANUAL

INTERNET OF THINGS (IoT) DEVICE ENGINEERING BASE ON ESP8266 (NodeMCU)



DADAN NURDIN BAGENDA, ST., MT.

TELKOM PCC

2024

47DAFTAR ISI

Konten	Hal
DAFTAR ISI	iii
DAFTAR GAMBAR	v
DAFTAR TABEL	vii
Part I. Mengenal Mikrokontroler dan IDE Arduino	1
I.1 Tujuan Workshop	1
I.2 Dasar Teori	1
I.1.1 Mikrokontroler	1
I.2.1 IDE Arduino	6
I.3 Alat dan Komponen yang Digunakan	7
I.4 Langkah-Langkah Percobaan	7
I.4.1 Langkah Persiapan	7
I.4.2 Langkah Workshop	11
I.5 Diskusi	13
Part II. ESP8266 Perangkat Komunikasi Serial, Wifi Lokal dan Online	14
II.1 Tujuan Workshop	14
II.2 Dasar Teori	14
II.3 Alat dan Komponen yang Digunakan	18
II.4 Langkah-Langkah Percobaan	18
II.4.1 Langkah Persiapan	18
II.4.2 Langkah Workshop	22
II.5 Diskusi	35
Part III. Input Analog	36
III.1 Tujuan Workshop	36

III.2 Dasar Teori	36
III.3 Alat dan Komponen yang Digunakan	39
III.4 Langkah-Langkah Percobaan	39
III.4.1 Langkah Persiapan.....	39
III.4.2 Langkah Workshop.....	40
Part IV. Sensor Jarak dan Output Analog.....	46
IV.1 Tujuan Workshop	46
IV.2 Dasar Teori.....	46
IV.3 Alat dan Komponen yang Digunakan	51
IV.4 Langkah-Langkah Percobaan	51
IV.4.1 Langkah Persiapan.....	51
IV.4.2 Langkah Workshop.....	52

LAMPIRAN

Lampiran A

DAFTAR GAMBAR

Gambar	Hal
Gambar I.1 Mikrokontroler, ESP8266, ESP32, ESP01, Arduino Nano, Due, Uno 3	
Gambar I.2 Pinout NodeMCU ESP8266 V3 Lolín.....	5
Gambar I.3 Foto NodeMCU ESP8266 V3 Lolín.....	5
Gambar I.4. Tampilan dari Software Arduino IDE.....	7
Gambar I.5 Preference IDE Arduino	8
Gambar I.6 Menambahkan Link Additional Boards Manager URL.....	8
Gambar I.7 Install Board ESP8266.....	9
Gambar I.8 Membuka Device Manager.....	10
Gambar I.9 CH340 sudah terinstal dan melihat nomor port COM.....	10
Gambar I.10 Mikrokontroler NodeMCU ESP8266 terhubung ke PC	11
Gambar I.11 Lampu LED_BUILTIN aktif.....	12
Gambar I.12 Menghubungkan kabel Female-Female dari pin D0 ke Gnd.....	12
Gambar II.1. Contoh komponen-komponen Input dan Output Mikrokontroler ...	15
Gambar II.2 Foto Modul Sensor LDR	17
Gambar II.3 Sign Up akun Ubidots.....	19
Gambar II.4 Akun Free Platform Ubidots.....	19
Gambar II.5 Create Ubidots Stem Account	19
Gambar II.6 Mengisi username dan password akun ubidots stem.....	20
Gambar II.7 Konfirmasi ubidots account melalui inbox email.....	20
Gambar II.8 Login di ubidots.com.....	20
Gambar II.9 Cari, install dan buka App Ubidots	21
Gambar II.10 Login aplikasi Ubidots IoT.....	22
Gambar II.11 Sensor LDR dan Buzzer ke NodeMCU ESP8266.....	23
Gambar II.12 Install Library Ubidots MQTT for ESP8266.....	26
Gambar III.1. Contoh algoritma percabangan.....	36
Gambar III.2. Contoh algoritma perulangan.....	37
Gambar III.3 Potensiometer	37
Gambar III.4 Relay 1 Channel	38
Gambar III.5 Pulse Width Modulation.....	39

Gambar IV.1 DHT11	46
Gambar IV.2 Ilustrasi Perhitungan Sensor Ultrasonik.....	48
Gambar IV.3 Sensor Ultrasonic	48
Gambar IV.4. Konstruksi motor servo.....	50
Gambar IV.5 Tower Pro Micro Servo MG90S.....	50

DAFTAR TABEL

Tabel	Hal
Tabel I.1 Pin I/O NodeMCU ESP8266 yang memiliki Internal PullUP Resistor...	6
Tabel II.1 Sensor LDR dan Buzzer ke NodeMCU ESP8266.....	22

Part I. Mengenal Mikrokontroler dan IDE Arduino

I.1 Tujuan Workshop

1. Peserta mengenal Mikrokontroler ESP8266 sebagai pengendali utama sistem embedded.
2. Peserta mampu menggunakan IDE Arduino sebagai Aplikasi Editor dan Compiler Arduino.

I.2 Dasar Teori

I.1.1 Mikrokontroler

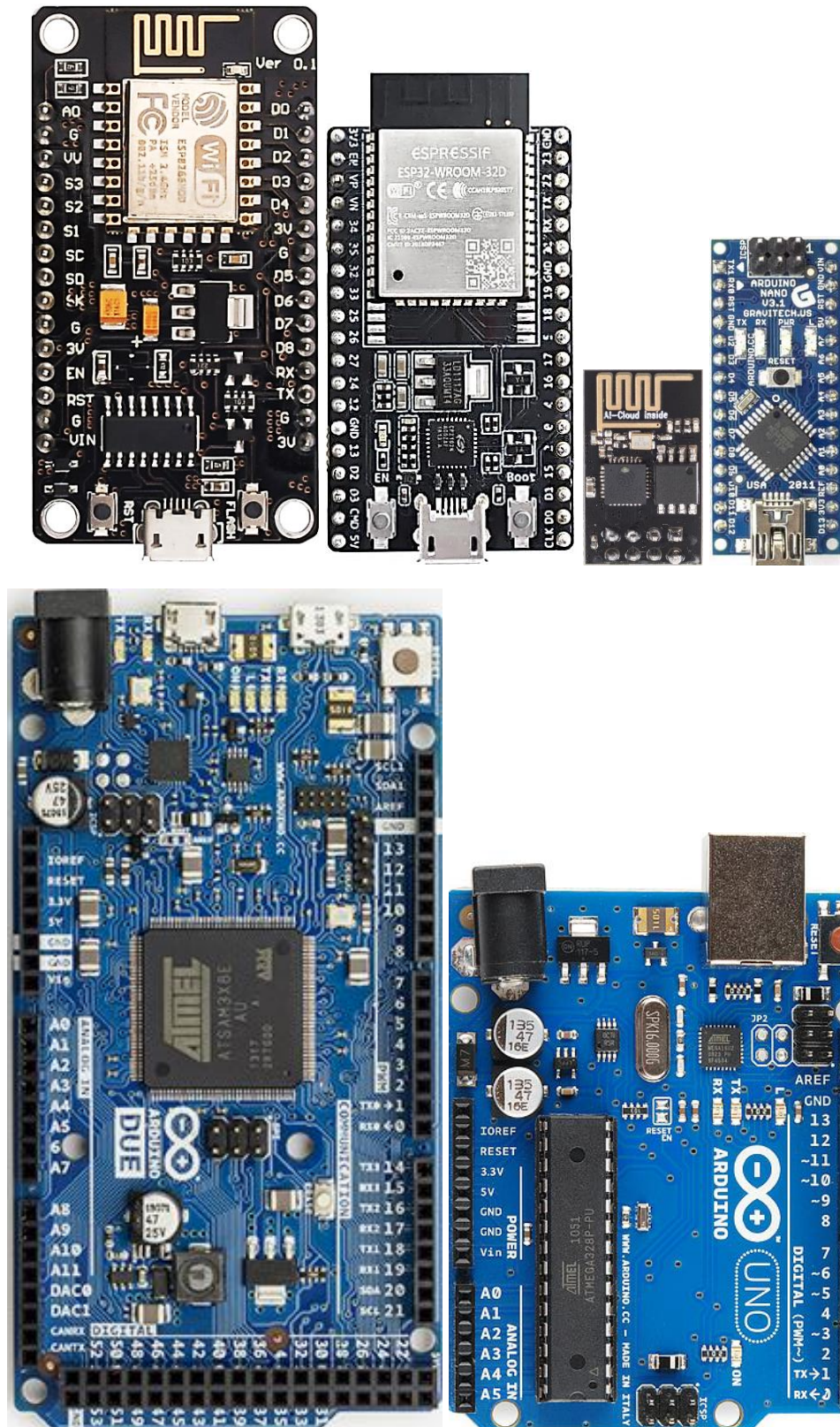
Mikrokontroler adalah suatu jenis integrated circuit (IC) yang mengintegrasikan unit pemrosesan pusat, memori, dan perangkat masukan/keluaran ke dalam satu chip kecil. Fungsi utama mikrokontroler adalah untuk mengontrol operasi dari suatu sistem atau perangkat elektronik. Mikrokontroler banyak digunakan dalam berbagai aplikasi, mulai dari perangkat rumah tangga hingga sistem kontrol industri.

Beberapa karakteristik umum dari mikrokontroler meliputi:

1. Unit Pemrosesan Pusat (CPU): Mikrokontroler memiliki CPU yang dapat melakukan berbagai operasi aritmetika dan logika. Ini adalah otak dari mikrokontroler yang mengatur eksekusi program.
2. Memori: Terdapat dua jenis memori utama pada mikrokontroler, yaitu program memory (tempat penyimpanan program yang akan dieksekusi) dan data memory (tempat penyimpanan variabel dan data sementara selama eksekusi program).
3. Perangkat Masukan/Keluaran (I/O): Mikrokontroler memiliki pin-ping I/O yang memungkinkan komunikasi dengan dunia luar, seperti sensor dan aktuator. Ini memungkinkan mikrokontroler untuk menerima informasi dari lingkungannya dan mengontrol perangkat eksternal.
4. Sistem Pengaturan Clock: Sebagian besar mikrokontroler memiliki sumber clock internal atau eksternal untuk mengatur kecepatan operasionalnya.

5. Peripheral Devices: Beberapa mikrokontroler memiliki perangkat tambahan seperti timer, counter, USART (Universal Synchronous/Asynchronous Receiver/Transmitter), PWM (Pulse Width Modulation), dan lainnya, yang memperluas kemampuan mikrokontroler untuk berbagai aplikasi.

Kelebihan penggunaan mikrokontroler termasuk ukuran yang kecil, konsumsi daya yang rendah, dan fleksibilitas yang tinggi dalam pengembangan berbagai sistem elektronik. Mikrokontroler banyak digunakan dalam perangkat seperti mikrooven, mesin cuci, kamera digital, sistem kontrol otomotif, dan sebagainya.



Gambar I.1 Mikrokontroler, ESP8266, ESP32, ESP01, Arduino Nano, Due, Uno

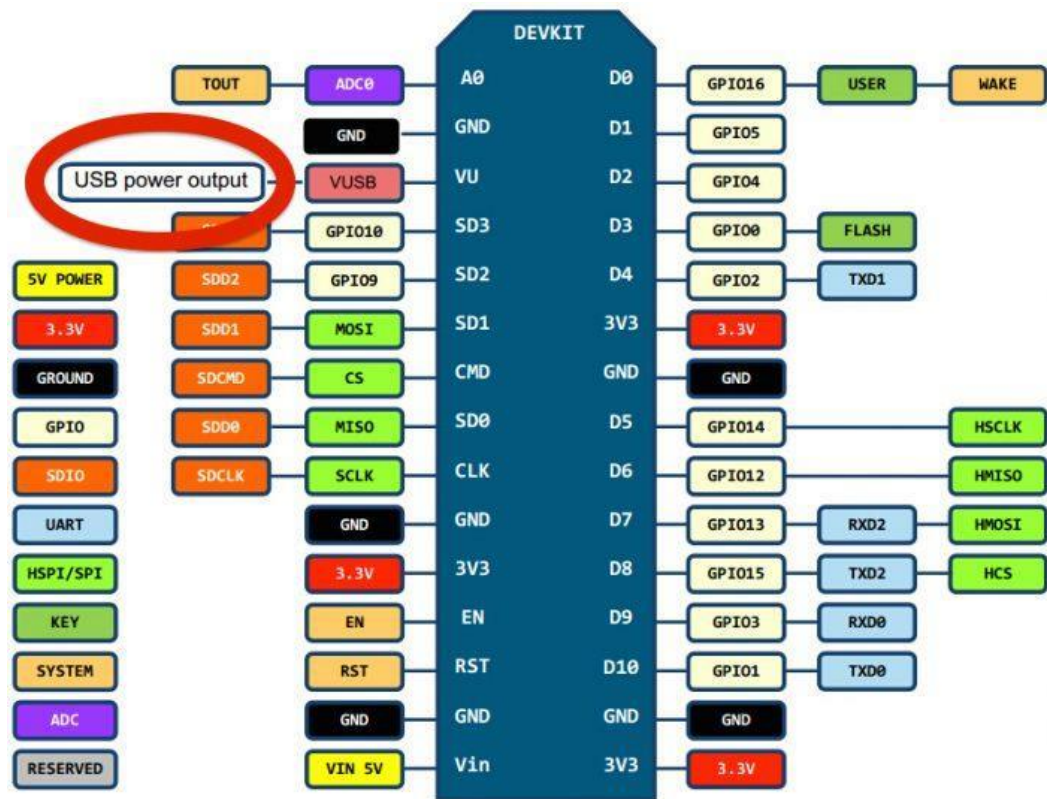
Ada banyak jenis mikrokontroler yang berbeda di pasaran, dan perbedaan antara mereka terletak pada arsitektur, fitur, kinerja, dan aplikasi yang ditujukan. Beberapa merek dan keluarga mikrokontroler yang populer termasuk:

1. Espressif Systems; Sebuah perusahaan yang terkenal dengan pengembangan mikrokontroler dan modul Wi-Fi yang populer dalam proyek Internet of Things (IoT), dengan fitur builtin Wi-Fi 802.11 b/g/n., UART
 - ESP8266 ; CPU Tensilica L106, clock hingga 160 MHz,
 - ESP32 ; penerus ESP8266, dua core prosesor Xtensa 32-bit LX6, Bluetooth Low Energy (BLE), SPI, I2C, clock hingga 240 MHz
2. Arduino; adalah platform pengembangan perangkat keras open-source yang mencakup papan mikrokontroler dan lingkungan pengembangan perangkat lunak (IDE). Platform ini dirancang untuk memudahkan orang yang tidak memiliki latar belakang teknik elektronik dan pemrograman yang mendalam untuk memulai dan mengembangkan proyek elektronik. Beberapa jenis board Arduino; Uno, Nano, Micro, Mega, Due, Leonardo.
3. Atmel AVR: Mikrokontroler AVR dikenal karena kepopulerannya dalam dunia hobi dan DIY. Mereka digunakan dalam Arduino, sebuah platform pengembangan elektronik open-source yang sangat populer.
4. ARM Cortex-M: Keluarga mikrokontroler ARM Cortex-M memiliki arsitektur yang lebih kompleks dan biasanya digunakan dalam aplikasi yang memerlukan kinerja tinggi. Banyak produsen mikrokontroler, seperti STMicroelectronics, NXP, dan Texas Instruments, memproduksi mikrokontroler berbasis arsitektur ARM Cortex-M.
5. Microchip PIC (Peripheral Interface Controller): PIC adalah keluarga mikrokontroler yang banyak digunakan dalam berbagai aplikasi. Mereka dikenal dengan arsitektur RISC (Reduced Instruction Set Computing) mereka dan dapat ditemukan dalam berbagai varian dengan berbagai fitur.
6. Raspberry Pi (bukan mikrokontroler, tapi single-board computer): Raspberry Pi adalah mini komputer, berjalan dengan Operating System Linux, papan tunggal yang memiliki kemampuan komputasi yang lebih besar dibandingkan dengan mikrokontroler.

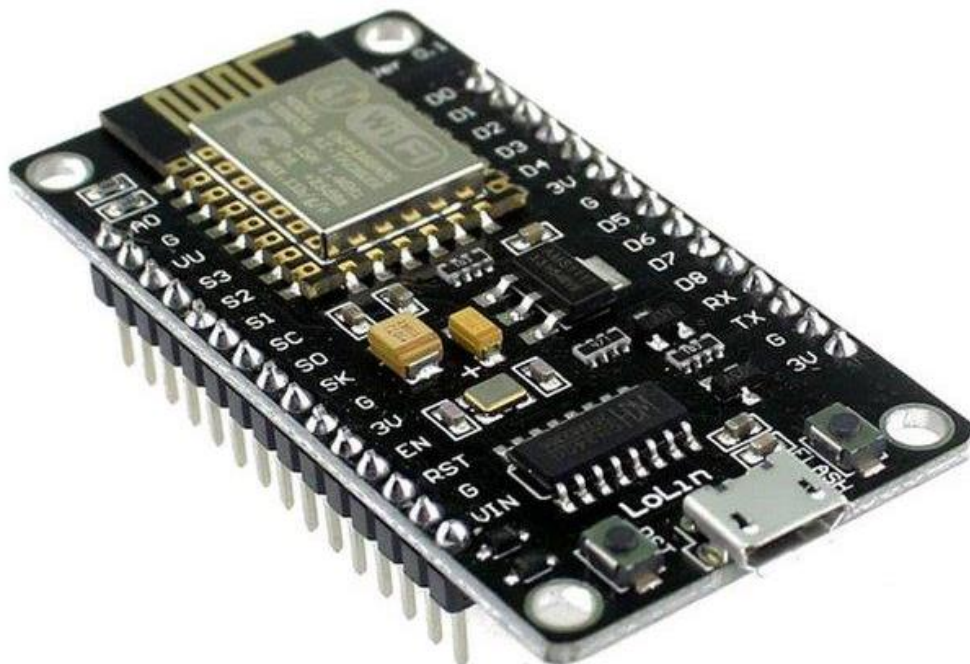
Perbedaan antar mikrokontroler dapat melibatkan hal-hal seperti arsitektur, kecepatan clock, kapasitas memori, jumlah pin I/O, dan kemampuan perangkat keras tambahan. Pemilihan mikrokontroler yang tepat tergantung pada persyaratan spesifik dari proyek atau aplikasi yang sedang dikembangkan. Beberapa mikrokontroler dirancang untuk aplikasi tertentu, seperti kendali motor, sistem sensor, atau komunikasi nirkabel, sementara yang lainnya dirancang untuk tugas umum.

Pada kelas ini kita akan gunakan mikrokontoler ESP8266 dengan operasi kerja tegangan JEDEC (tegangan 3.3VDC), Tidak seperti mikrokontroler AVR dan sebagian besar board Arduino yang memiliki tegangan TTL 5 volt. Meskipun begitu, node mcu masih bisa terhubung dengan 5V namun melalui port micro USB atau pin Vin yang disediakan oleh board-nya. Namun karena semua pin pada ESP8266 tidak toleran terhadap masukan 5V. Maka sebaiknya tidak langsung

mencatunya dengan tegangan TTL jika tidak ingin merusak board anda. Anda bisa menggunakan Level Logic Converter untuk mengubah tegangan ke nilai aman 3.3v. ESP8266 memiliki pin Input Output sbb.:



Gambar I.2 Pinout NodeMCU ESP8266 V3 Lolín



Gambar I.3 Foto NodeMCU ESP8266 V3 Lolín

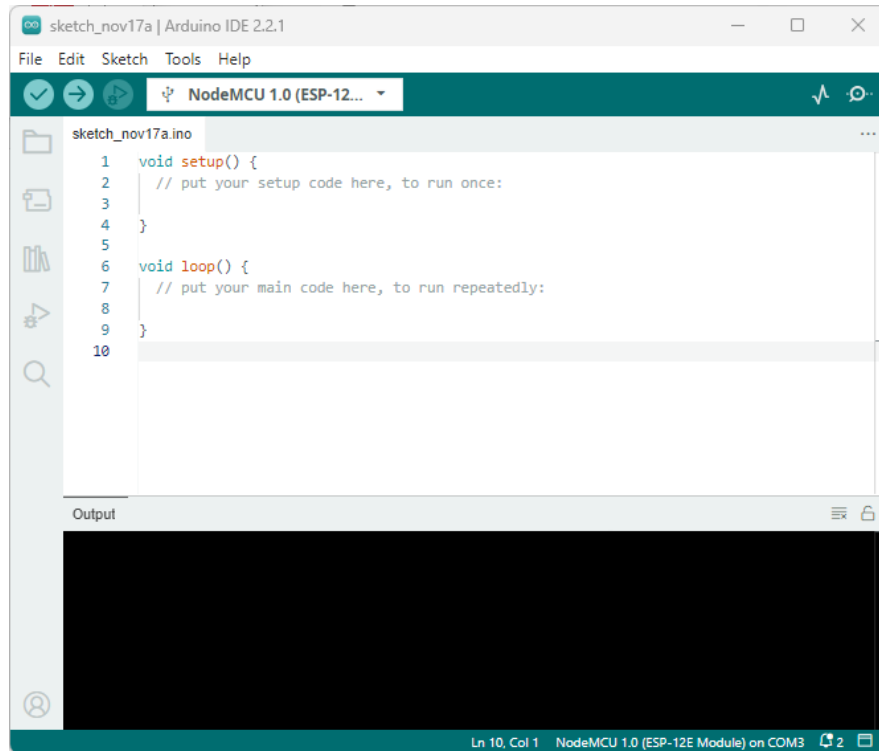
Tabel I.1 Pin I/O NodeMCU ESP8266 yang memiliki Internal PullUP Resistor

Pin	Name	Desired level during power on	Before first software download	Necessary Function	ANT Switch	Extendable function	Extendable GPIO/I2C
8	XPD_DCDC		Output high	Deep-Sleep Wakeup	Yes [0]		GPIO16 (optional internal pull-down)
9	MTMS		Input, Internal pull-up resistor			HSPICLK	GPIO14 (optional internal pull-up)
10	MTDI		Input, Internal pull-up resistor			HSPIQ	GPIO12 (optional internal pull-up)
12	MTCK		Input, Internal pull-up resistor			HSPID	GPIO13 (optional internal pull-up)
13	MTDO	Low	Input, Internal pull-up resistor			HSPICS	GPIO15 (optional internal pull-up)
14	GPIO2	High	Output, Many toggles	U1TXD			GPIO2 (optional internal pull-up)
15	GPIO0	High (uart download) Low	Output, Many toggles		Yes [0]	SPICS2	GPIO0 (optional internal pull-up)
16	GPIO4		Input, Internal pull-up resistor				GPIO4 (optional internal pull-up)
18	SD_DATA_2		Input, Hi-Z	SPIHD	Yes	HSPiHD	GPIO9 (optional internal pull-up)
19	SD_DATA_3		Input, Hi-Z	SPIWP	Yes[0]	HSPiWP	GPIO10 (optional internal pull-up)
20	SD_CMD		Input, Hi-Z	SPICSS0			No
21	SD_CLK		Input, Hi-Z	SPICLK			No
22	SD_DATA_0		Input, Hi-Z	SPIQ			No
23	SD_DATA_1		Input, Hi-Z	SPID	Yes		No
24	GPIO5		Input, Internal pull-up resistor				GPIO5 (optional internal pull-up)
25	U0RXD		Input, Internal pull-up resistor	U0RXD	Yes		GPIO3 (optional internal pull-up)
26	U0TXD	High	Output, Many toggles	U0TXD	Yes	SPICS1	GPIO1 (optional internal pull-up)

I.2.1 IDE Arduino

IDE itu merupakan kependekan dari Integrated Development Environment, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang diberikan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (Sketch) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama Bootlader yang berfungsi sebagai penengah antara compiler Arduino dengan mikrokontroler.

IDE Arduino memiliki tampilan program editor seperti Gambar I.4 yang berfungsi sebagai Program Editor, Compiler, Uploader, Monitor dan Plotter. Compiler ini mengubah file program *.ino menjadi *.asm *.hex dan lainnya, hex file yang terbentuk akan kita tanamkan pada Arduino Board.



Gambar I.4. Tampilan dari Software Arduino IDE

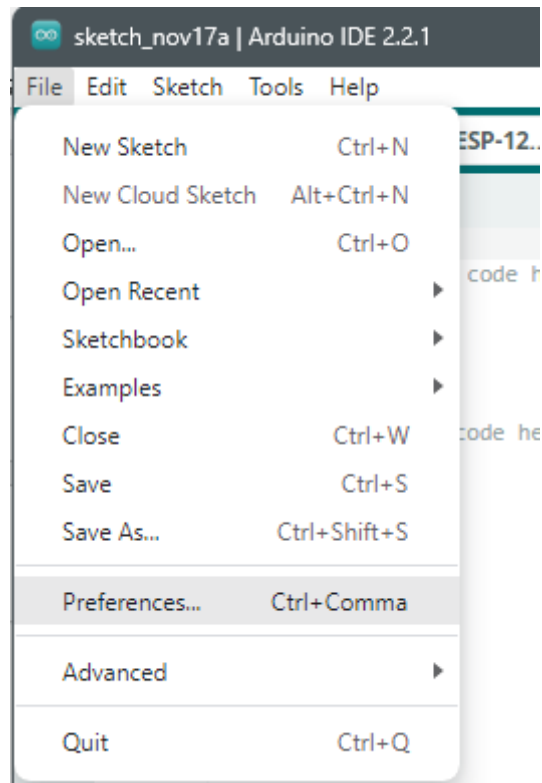
I.3 Alat dan Komponen yang Digunakan

- 1) Mikrokontroler NodeMCU ESP8266
- 2) Kabel data USB to Micro USB
- 3) Laptop/PC OS Win 7-10 terhubung ke internet
- 4) Software IDE Arduino
- 5) Driver CH341SER.exe
- 6) Wifi Online

I.4 Langkah-Langkah Percobaan

I.4.1 Langkah Persiapan

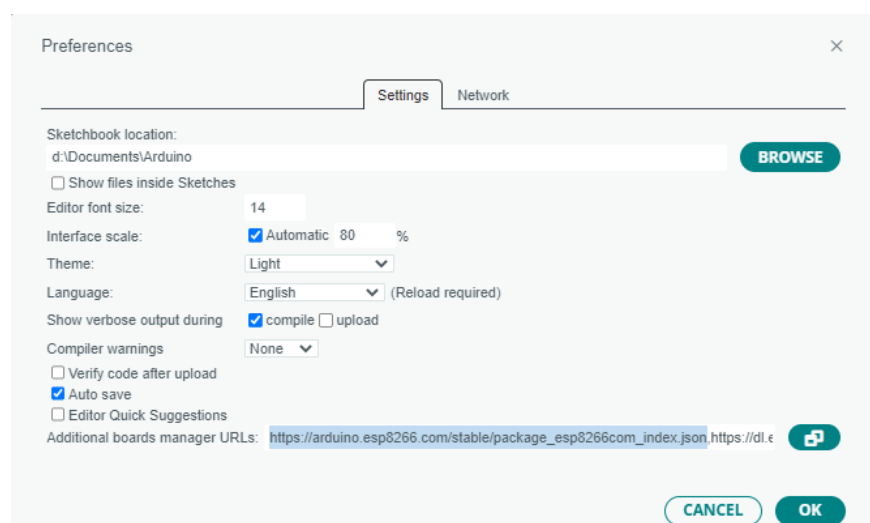
- 1) Install IDE Arduino lastest version pada Laptop/PC yang digunakan. Jika belum memiliki installernya silahkan download di web www.arduino.cc, tab software -> Windows Win 10 and newer, 64 bits (menyesuaikan)
- 2) Pada IDE Arduino, instal board sekaligus library NodeMCU ESP8266



Gambar I.5 Preference IDE Arduino

- a. File->Preference->”Additional Boards Manager URLs:” Isikan atau tambahkan link berikut :

https://arduino.esp8266.com/stable/package_esp8266com_index.json

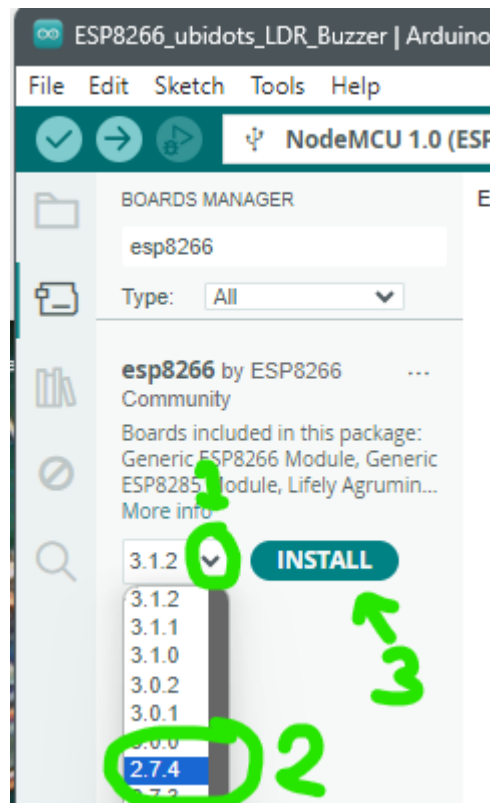


Gambar I.6 Menambahkan Link Additional Boards Manager URL

- b. Langkah berikutnya laptop **harus Online**;

Tools->Board->Board Manager

Jika sudah terbuka ketik “ESP8266”, Selanjutnya **pilih versi 2.7.4**, click install dan tunggu sampai selesai, seperti gambar di bawah Pilih Install. Jika sudah muncul pesan “Installed”, bisa di close manual dan buka kembali. Ini cukup lama tergantung kecepatan internet.



Gambar I.7 Install Board ESP8266

- c. Jika sudah terinstal, pada aplikasi Arduino IDE tab “Tools” pilih:

Tools->Board->ESP8266 board->NodeMCU1.0(ESP-12E Module)


- d. Jika masih gagal coba link lain seperti informasi pada web <https://arduino-esp8266.readthedocs.io/en/latest/>

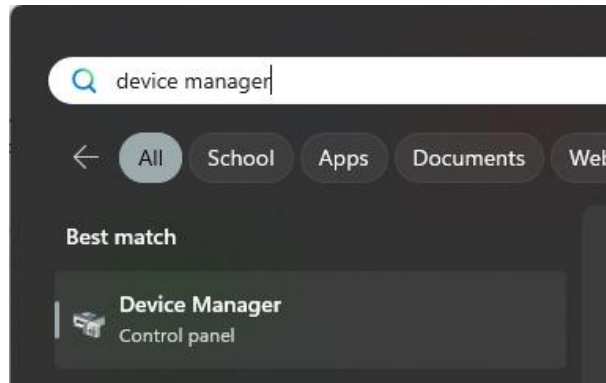
3) Hubungkan NodeMCU ESP8266 pada PC menggunakan kabel Micro USB.

4) Install CH341SER.exe sebagai driver NodeMCU ESP8266 Lolin, , download pada link berikut:

<https://drive.google.com/file/d/1K47-16WYSCatENNoRETVycD7jodXesp0/view?usp=sharing>

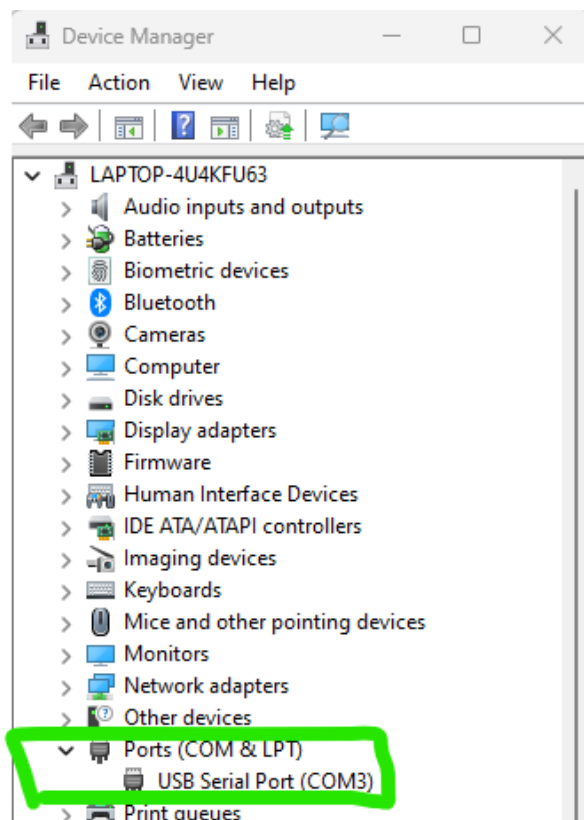
pastikan sudah terinstal dengan baik, lihat pada “Device Manager”;

- a. Buka “Device Manager”, dengan mengetiknya di jendela pencarian windows. Tekan tombol , lalu ketikkan “Device Manager”



Gambar I.8 Membuka Device Manager

- b. lihat pada bagian “Ports (COM & LPT)” dan ingat **nomor COM** CH340-nya



Gambar I.9 CH340 sudah terinstal dan melihat nomor port COM

- c. Jika belum berhasil silahkan ulangi dari nomor 3 lalu 4a b c, atau silahkan menghubungi instruktur.
- 5) Ubah parameter board di IDE dari Arduino Uno menjadi NodeMCU

- a. Tools->Board->ESP8266 Boards->NodeMCU 1.0 (ESP-12e)
 - b. Tools->Port:COMX, nomor COM sesuaikan dengan yang dilihat pada Device Manager.
- 6) Siapkan nama SSID dan Password dari Access Point/Hotspot yang tersedia.

I.4.2 Langkah Workshop

I.4.2.1 Output Digital 1 Bit, Blink pada lampu LED_BUILTIN

1. Hubungkan Mikrokontroler NodeMCU ESP8266 dengan PC menggunakan kabel Micro USB. Tidak ada rangkaian tambahan, karena menggunakan LED_BUILTIN (sebenarnya itu Led yang terhubung ke pin D4 sebagai indikator TX komunikasi WiFi).



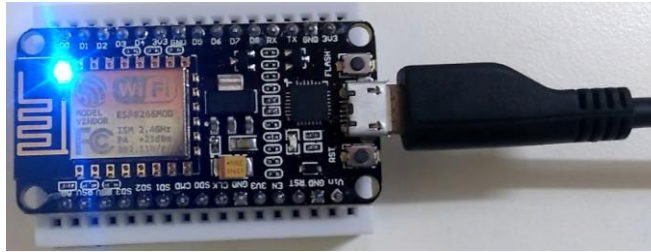
Gambar I.10 Mikrokontroler NodeMCU ESP8266 terhubung ke PC

2. Pada IDE arduino Buka File->Examples->ESP8266->Blink. Atau masukan program berikut, simpan dengan nama ESP8266_Blink:

```
/*
D0=16, D1=5, D2=4, D3=9, D4=2=Tx1=Led_activeLow, D5=14,
D6=13, D7=13=Rx2, D8=15=Tx2, Rx0=D9=3, Tx0=D10=1
*/
void setup() {
  pinMode(D4,OUTPUT); //setting D4 sebagai output
}
void loop() {
  digitalWrite(D4,0); //On selama 100ms
  delay(100);
  digitalWrite(D4,1); //Off selama 1000ms
  delay(1000);
}
```

3. Masukan program ke Mikrokontroler ESP8266 dengan menekan CTRL+U. Lihat pada bagian halaman OUTPUT IDE Arduino, akan muncul Compiling, lalu Uploading, saat uploading tertulis pada console “Conecting, Writing lalu Hard resetting” artinya program berhasil tertanam.
4. Testing dan Observasi. Perhatikan dan pahami setiap baris programnya,

- LED Biru dekat antena akan berkedip sesuai program.

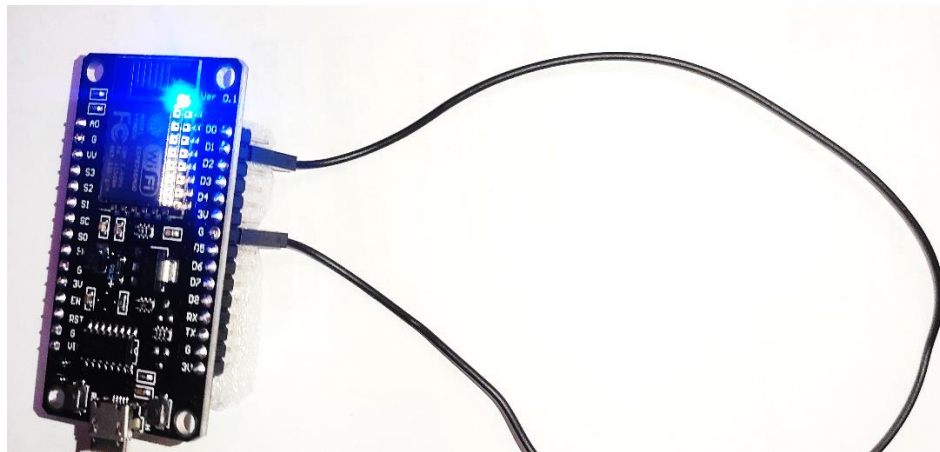


Gambar I.11 Lampu LED_BUILTIN aktif

- Karena Active LOW, sehingga ON akan lebih sebentar dari OFF sesuai Program.
5. Pengembangan kasus, coba dibalik On 2 dan OFF 1/2 detik, dan upload kembali. Jika berhasil raise hand dan tunjukan pada instruktur

I.4.2.2 Membaca Input Digital pada pin Mikrokontroler ESP8266

1. Hubungkan 1 kabel jumper Female-Female dari pin D1 ke G atau ground, jumper ini berperan sebagai saklar.



Gambar I.12 Menghubungkan kabel Female-Female dari pin D1 ke Gnd

2. Buat file baru pada IDE Arduino, dan simpan dengan nama ESP8266_InputDigital
3. Ketik dan Upload code program berikut:

```
int baca; //buat variable baca
void setup() {
  pinMode(D4, OUTPUT); //Set Pin D4 sbg Output
  pinMode(D1, INPUT_PULLUP); //Set Pin D1 sbg Input
}
```

```
void loop() {  
  baca=digitalRead(D1); //lihat logic di pin-D1 hasilnya  
  pada baca  
  if (baca==0){ //jika terbaca 0 artinya tombol ditekan  
    digitalWrite(D4, LOW ); //hidupkan LED  
  } //jika tidak 0 artinya tombol dilepas  
  else{ //selain itu  
    digitalWrite(D4, HIGH); //matikan LED  
  }  
}
```

4. Testing dan Observasi

Pada langkah ini seharusnya;

- Saat button ditekan atau Pin D1 dihubungkan ke Gnd, maka lampu LED On,
- Sedangkan saat button dilepas atau Pin D1 dilepas, maka lampu LED Off.

5. Pengembangan kasus, ubahlah program sehingga

- jika button ditekan maka LED berkedip cepat,
- jika button dilepas maka LED mati.

I.5 Diskusi

Part II. ESP8266 Perangkat Komunikasi Serial, Wifi Lokal dan Online

II.1 Tujuan Workshop

1. Peserta mampu menggunakan LDR sebagai Sensor Cahaya Tampak menjadi input bagi mikrokontroler
2. Peserta mampu menggunakan buzzer sebagai alarm menjadi output bagi mikrokontroler
3. Peserta mampu menggunakan mikrokontroler NodeMCU ESP8266 untuk berkomunikasi dengan sinyal Serial Input dan Serial Output.
4. Peserta mampu menggunakan mikrokontroler NodeMCU ESP8266 menjadi web server yang diakses secara lokal melalui wifi untuk pengendalian lampu LED dan monitoring Digital.
5. Peserta mampu menggunakan mikrokontroler NodeMCU ESP8266 menjadi client IoT Device yang diakses secara Online melalui Internet web server Ubidots untuk pengendalian digital (LED/Buzzer) dan monitoring digital (Button/Sensor LDR).

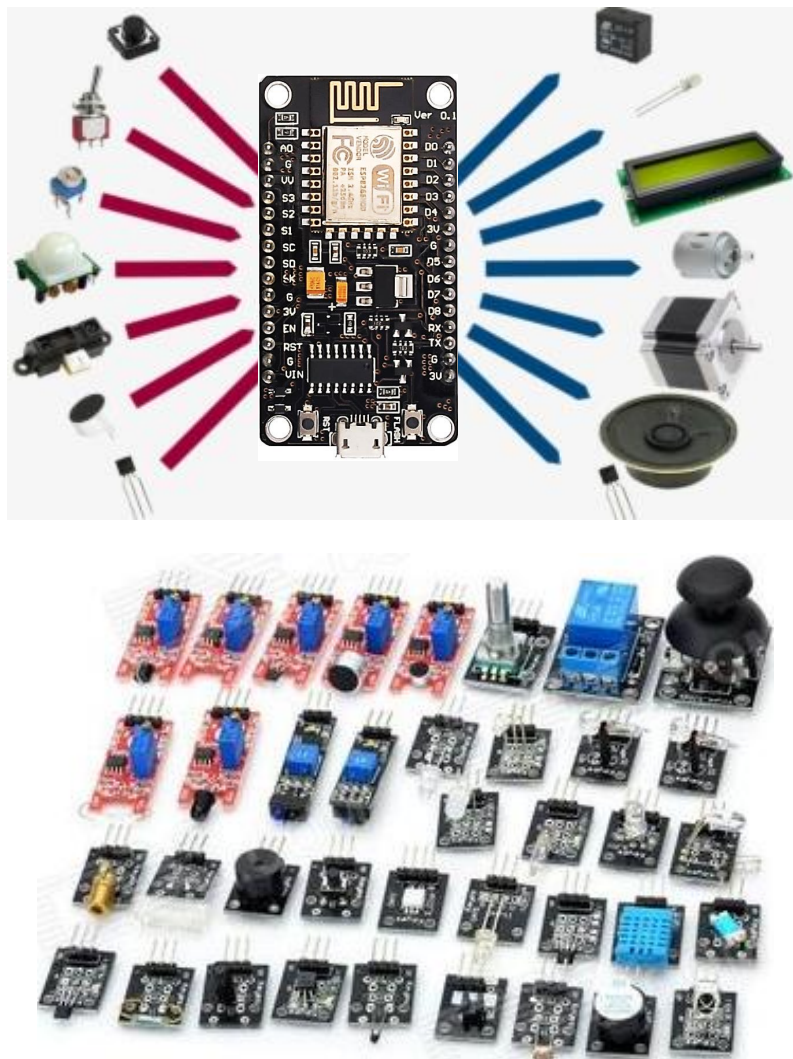
II.2 Dasar Teori

ESP8266 adalah mikrokontroler yang memiliki Wi-Fi built-in yang banyak digunakan untuk proyek IoT. Part I sebelumnya kita menggunakan Input digital (button) dan Output digital (Led) secara langsung pada pin NodeMCU ESP8266, pada Part II sekarang kita awali dengan mengakses kemampuannya berkomunikasi melalui sinyal Serial Input dan Serial Output.

Sinyal Serial adalah sinyal digital lebih dari 1 bit yang setiap bit-nya dikirim satu-persatu bergiliran, ini adalah pengembangan dari pengiriman sinyal yang sebelumnya paralel yang membutuhkan jalur transmisi lebih banyak. Sinyal Serial ini bisa berupa listrik cahaya maupun suara seperti morse. Sinyal serial ini banyak macamnya dengan berbeda-beda protokol (nama, timing, level, bits dsb.). Contohnya yang kita gunakan sekarang Universal Synchronous Asynchronous Receiver Transmitter (USART), 9600 bps, 8 data, 2 stop bits. Protokol ini banyak digunakan, sehingga memungkinkan Arduino dapat berkomunikasi dengan banyak

device dengan protokol yang sama. Contoh protokol serial lainnya; USB, PS2, TCP/IP dsb. **Fungsi Serial built-in** dari NodeMCU ESP8266 bisa diakses melalui socket microUSB-nya, juga bisa melalui pin RX dan TX secara langsung seperti yang ditunjukkan pada Gambar I.2 Pinout NodeMCU ESP8266 V3 Lolin.

Device Input salahsatunya **Sensor** merupakan komponen/Hardware yang memberikan masukan signal pada Mikrokontoler, pada umumnya dari besaran **fisis ke besaran listrik**, contohnya: Button, Potensiometer, Sensor dsb. Sedangkan **Device Output** juga identik dengan **Aktuator**, merupakan komponen/Hardware yang menerima signal dari Mikrokontoler, pada umumnya dari besaran besaran listrik menjadi besaran fisis, contohnya: LED, Buzzer, Motor DC, Heater dsb. Contoh Input dan Output lainnya dapat dilihat pada gambar II.1.



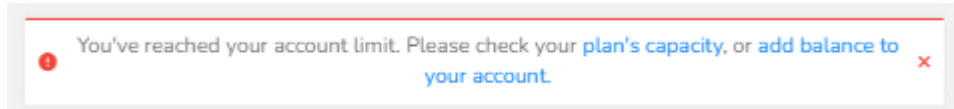
Gambar II.1. Contoh komponen-komponen Input dan Output Mikrokontroler

Web Server adalah hardware sekaligus software yang menyediakan layanan untuk mengakses **halaman web** atau sumber daya lain melalui jaringan, seperti internet(Online) atau intranet(lokal). Ini memungkinkan pengguna atau klien untuk mengakses dan menampilkan informasi yang dihosting oleh server baik secara lokal maupun online melalui internet. **Web Server NodeMCU ESP8266** bisa diaktifkan dengan menanamkan programnya dan dihubungkan ke jaringan melalui Wifi, lalu halamannya bisa diakses oleh smartphone ataupun laptop, dari halaman ini kita bisa memonitor juga mengendalikan I/O hardwarenya. Namun karena keterbatasan memorynya yang kecil halaman web ini biasanya hanya berupa text tanpa image audio apalagi video.

NodeMCU ESP8266 bisa difungsikan sebagai Internet of Things (IoT) device, dimana alat ini menjadi client yang dihubungkan dengan Web Server secara Online melalui internet. Salah satu web server online yang dapat kita gunakan secara gratis adalah **Ubidots** sebagai platform IoT cloud. Agar NodeMCU ESP8266 berperan sebagai client IoT Device yang terhubung dengan server Ubidots, kita harus membuat akun Ubidots, NodeMCU harus diprogram terhubung ke hotspot (SSID dan password wifi), lalu menyesuaikan token nama device dan nama datanya ke Ubidots. Selanjutnya kita dapat mengendalikan (lampu LED) dan monitoring digital (button/sensor) secara online melalui Internet. Kapasitas maksimal dari akun Ubidots Stem sbb:

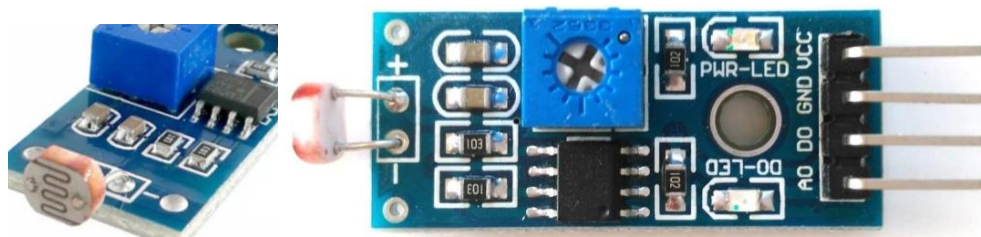
1. Perangkat: 3 perangkat gratis.
2. Dasbor: 1 dasbor, dengan hingga 10 widget.
3. Variabel: Hingga 10 variabel per perangkat.
4. Plugin: Hingga 1 plugin, dengan 10.000 eksekusi plugin per bulan.
5. Penerimaan Data: 4.000 titik per hari di seluruh perangkat Anda.
6. Ekstraksi Data: 500.000 titik per hari di seluruh akun Anda.
7. Laju Data: 1 permintaan per detik, di seluruh perangkat Anda.
8. Retensi Data: 1 bulan.
9. SMS & Panggilan Suara: 10 SMS dan 1 panggilan suara gratis per bulan (hanya untuk AS & Kanada), kemudian tarif standar berlaku.
10. Surel: 100 surel gratis per bulan, kemudian tarif standar berlaku.

11. Waktu Aktif: Tanpa SLA - sumber daya server bersama di antara semua pengguna STEM. Kecepatan dan keandalannya berdasarkan total permintaan platform pada setiap detik tertentu.
12. Dukungan: Berbasis komunitas, atau layanan mandiri menggunakan Pusat Bantuan dan Dokumentasi API kami.



Gambar pesan diatas contoh jika sudah melebihi kapasitas. Jika $24h=1440m=86400s$, dengan batasan kirim data ke Cloud 4000/hari maka delay program paling cepat 21s agar bisa 24 jam. Sedangkan delay 10s bisa 12jam, 5s bisa 6jam dst.

LDR atau Light Dependent Resistor adalah salah satu jenis sensor cahaya yang mengubah resistansinya berdasarkan intensitas cahaya yang diterimanya. Modul sensor LDR merupakan suatu paket yang mencakup LDR sebagai elemen sensor utama dan komponen tambahan untuk mempermudah penggunaan sensor tersebut, seperti yang ditunjukkan pada gambar berikut:



Gambar II.2 Foto Modul Sensor LDR

LDR (Light Dependent Resistor) Ini adalah komponen utama yang merespons perubahan intensitas cahaya dengan mengubah resistansinya. Pada kondisi gelap, resistansi LDR tinggi, sedangkan pada kondisi terang, resistansinya rendah. Trimport yang berwarna biru pada modul untuk mengatur sensitivitas sensor. Dengan memutar Trimport, pengguna dapat menyesuaikan ambang batas atau tingkat sensitivitas sensor terhadap perubahan cahaya. Modul sensor LDR biasanya memiliki output yang dapat dihubungkan ke mikrokontroler atau

perangkat lainnya. Output ini dapat berupa sinyal analog atau digital, tergantung pada desain modul.

Cara kerja modul sensor LDR saat cahaya mengenai LDR, tegangan pada pin A0 berubah (analog), nilai tegangan ini dapat digunakan untuk mengukur intensitas cahaya. Output dari pin D0 modul ini berupa digital dapat dibaca oleh mikrokontroler atau perangkat lain dalam bentuk logic untuk mengambil keputusan atau melakukan tindakan tertentu berdasarkan tingkat intensitas cahaya yang dideteksi, saat terang pin D0 ini mengeluarkan logic 0 dan saat gelap akan mengeluarkan logic 1. Aplikasi umum dari modul sensor LDR termasuk dalam proyek-proyek elektronika seperti lampu otomatis (yang menyala saat gelap), sistem pencahayaan otomatis, dan proyek-proyek berbasis Internet of Things (IoT) di mana pengukuran intensitas cahaya dapat digunakan untuk mengontrol perangkat lainnya.

II.3 Alat dan Komponen yang Digunakan

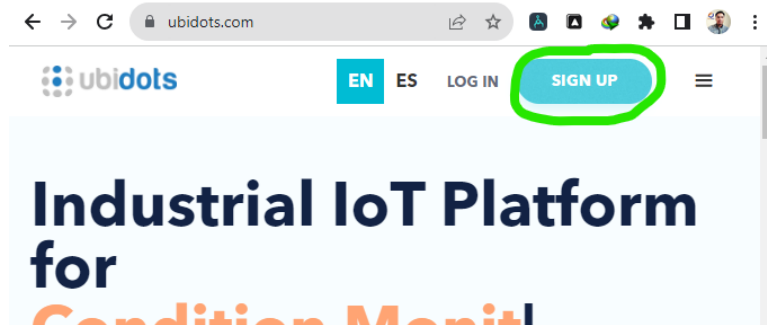
- 1) Laptop/PC dengan Software IDE Arduino latest version
- 2) NodeMCU ESP8266 Board + USB Cable 1 buah
- 3) Modul sensor cahaya nampak (LDR) 1 buah
- 4) Modul alarm (buzzer) 1 buah
- 5) Button/Kabel jumper female-female 3 buah + 3 buah
- 6) SmartPhone online + Apps freespace 200MB .

II.4 Langkah-Langkah Percobaan

II.4.1 Langkah Persiapan

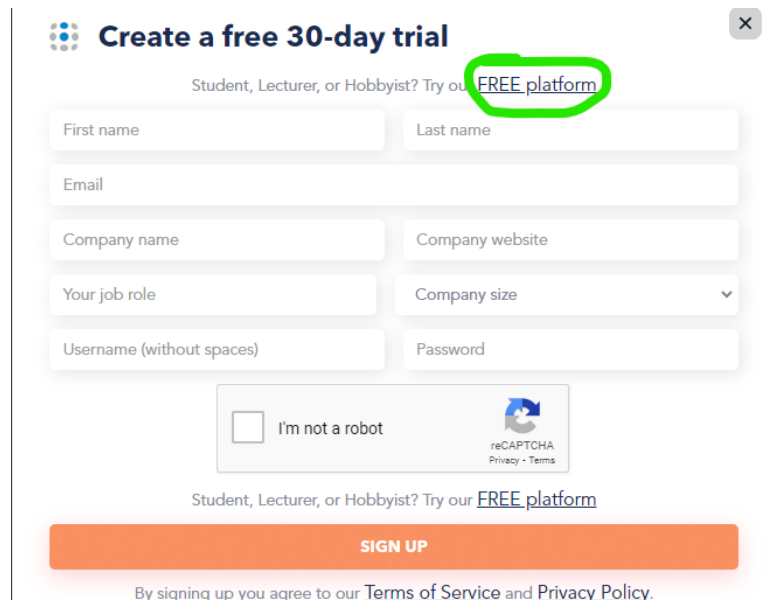
- 1) Tersedia IDE Arduino, pastikan sudah terinstal sudah terinstal ESP8266 board, pilih dengan cara:
Tools->Board->ESP8266 board->NodeMCU1.0(ESP-12E Module)
- 2) Tersedia Laptop/PC, pastikan hardware ESP8266 sudah terinstal. Buka “Device Manager” pada windows anda, dan lihat nomor COM yang terhubung.
- 3) Membuat akun Ubidots;

- a. <https://ubidots.com/>, pilih sign up



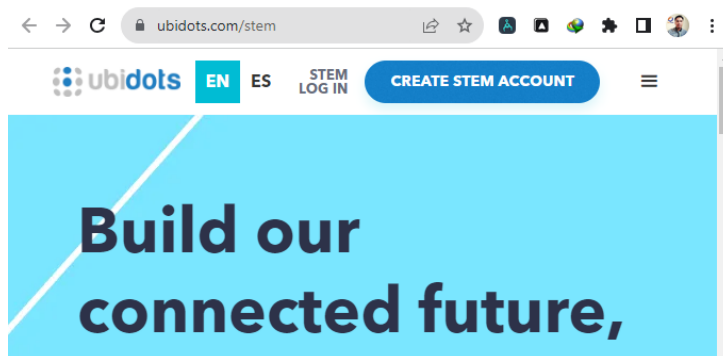
Gambar II.3 Sign Up akun Ubidots

- b. Pilih “FREE platform” untuk pembuatan akun edukasi, agar bisa digunakan lebih dari 1 bulan.



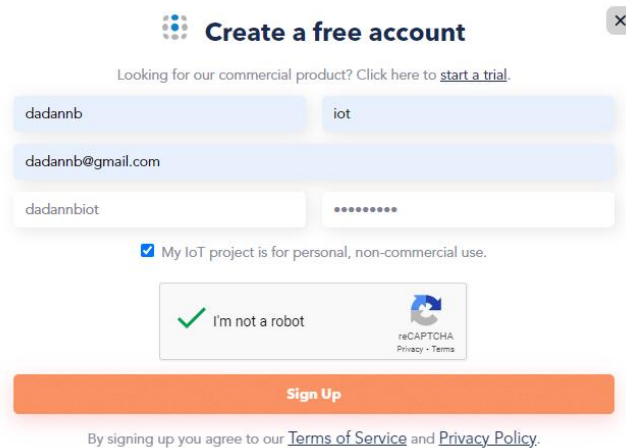
Gambar II.4 Akun Free Platform Ubidots

- c. Lanjutkan “Create Stem Account”



Gambar II.5 Create Ubidots Stem Account

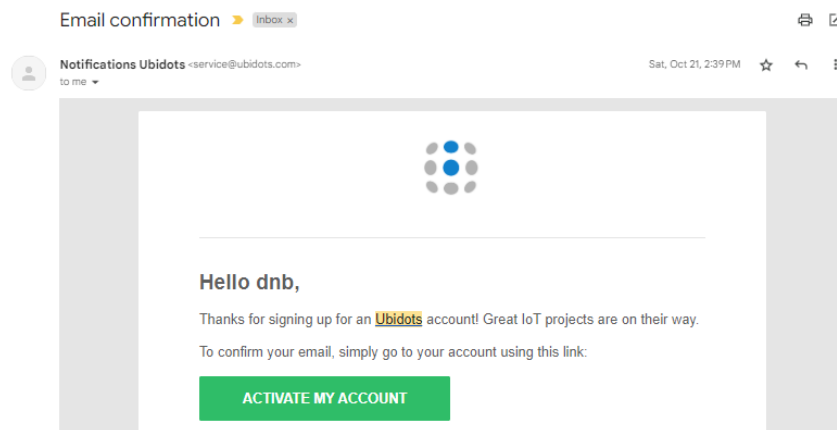
- d. Isi semua data, dicatat username dan password akun ubidots stem



The screenshot shows the 'Create a free account' page on Ubidots. It includes a close button (X) in the top right corner. Below the heading, there is a link: 'Looking for our commercial product? Click here to [start a trial](#).' The form contains several input fields: a username field with 'dadannb', a company field with 'iot', an email field with 'dadannb@gmail.com', and a password field with 'dadannbiot'. A checkbox is checked with the text 'My IoT project is for personal, non-commercial use.' Below the inputs is a reCAPTCHA widget with a green checkmark and the text 'I'm not a robot'. At the bottom is a large orange 'Sign Up' button. Below the button, it says 'By signing up you agree to our [Terms of Service](#) and [Privacy Policy](#).'

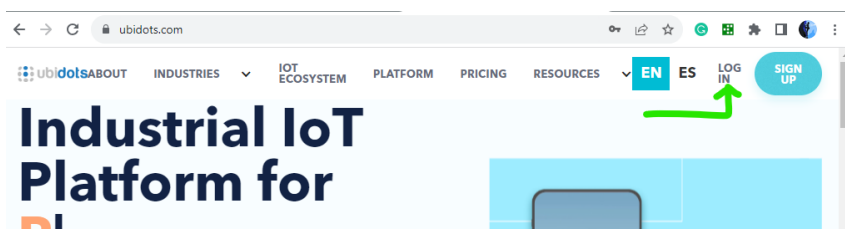
Gambar II.6 Mengisi username dan password akun ubidots stem

e. Konfirmasi ubidots account melalui inbox email yang didaftarkan



Gambar II.7 Konfirmasi ubidots account melalui inbox email

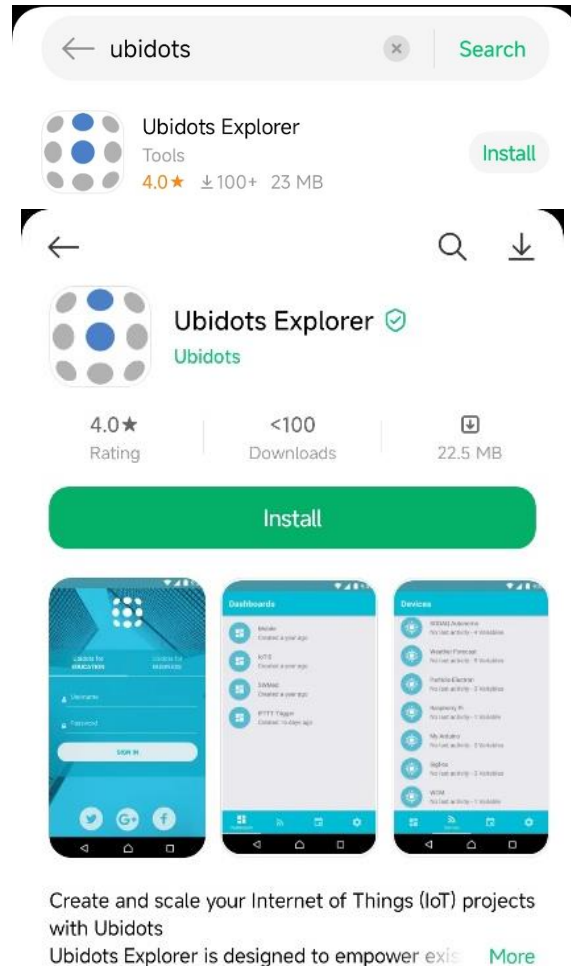
f. Login di ubidots.com



Gambar II.8 Login di ubidots.com

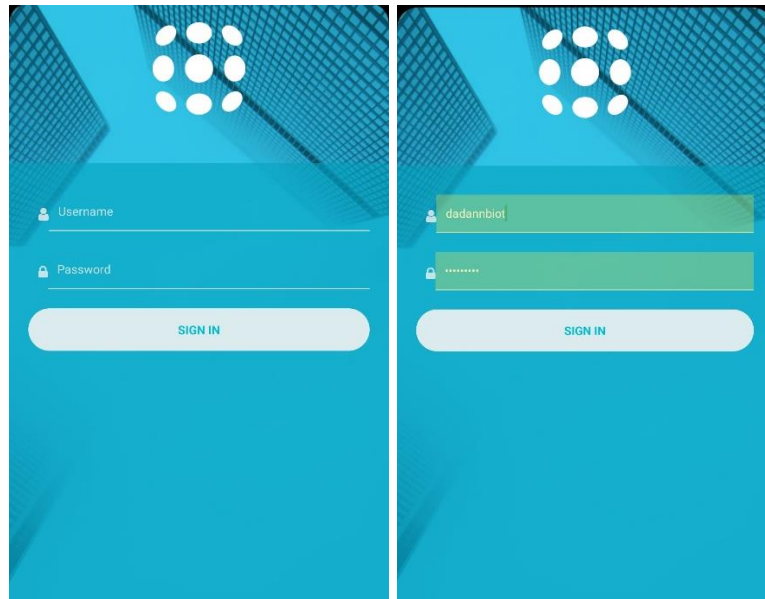
4) Install aplikasi “Ubidots Explorer” di Smartphone;

- a. Cari aplikasi “ubidots” dari **GetApps** atau **Play Store**.
Jika tidak ditemukan, silahkan download disini
https://drive.google.com/file/d/1OSy2WAjiK5RcXN0MS7zIXfft9_yp5mlE/view?usp=sharing
Jika tidak bisa juga, buka saja web ubidots pada web browser
smartphone Anda.



Gambar II.9 Cari, install dan buka App Ubidots

- b. Buka aplikasi Ubidots IoT dan lakukan Login akunnya



Gambar II.10 Login aplikasi Ubidots IoT

II.4.2 Langkah Workshop

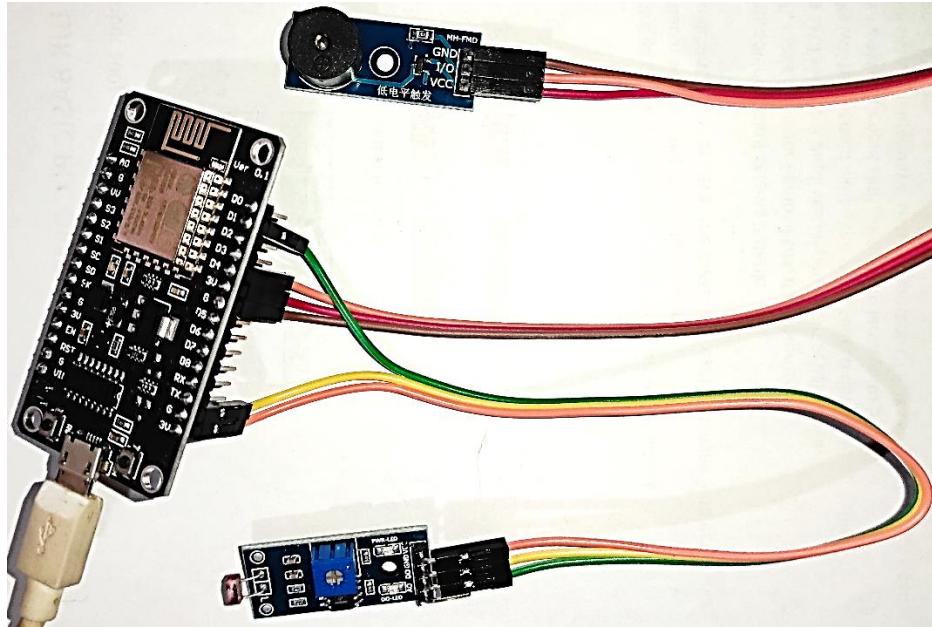
II.4.2.1 Menggunakan Sensor cahaya (LDR) dan alarm (Buzzer)

1. Hubungkan Sensor LDR dan Buzzer ke NodeMCU ESP8266

Tabel II.1 Sensor LDR dan Buzzer ke NodeMCU ESP8266

ESP8266	Sensor LDR
3V	Vcc
G	Gnd
D1	D0
X (abaikan)	A0

ESP8266	Buzzer
G	Gnd
D4	I/O
3V	Vcc



Gambar II.11 Sensor LDR dan Buzzer ke NodeMCU ESP8266

2. Buka program sebelumnya “ESP8266_InputDigital” dari percobaan I.4.2.2. lalu Save as “ESP8266_InputDigital_buzzer”, edit sbb;

```
int baca; //buat variable baca sensor
void setup() {
  pinMode(D4, OUTPUT); //Set Pin D4 sbg Output buzzer
  pinMode(D1, INPUT_PULLUP); //Set Pin D1 sbg Input LDR
}
void loop() {
  baca=digitalRead(D1); //baca pin-D1 simpan di var baca
  if (baca==0){ //jika var baca logic 0 artinya terang
    digitalWrite(D4, LOW); //hidupkan buzzer
  }
  else{ //selain itu //jika tidak logic 0 artinya gelap
    digitalWrite(D4, HIGH); //matikan buzzer
  }
}
```

3. Compile sekaligus tanamkan programnya. Sketch->upload atau "Ctrl+U"
4. Testing dan Observasi, pada langkah ini seharusnya:
 - Buzzer bersuara jika sensor LDR terang
 - Buzzer tidak bersuara jika sensor LDR gelap tertutup bayangan
 - NB: sensitifitas modul LDR terhadap cahaya bisa diatur pada trimpot
5. Pengembangan kasus: Bagaimana agar sistem bekerja dengan kondisi terbalik, alarm bersuara ketika terpicu oleh bayangan tamu tak diundang?

II.4.2.2 Kendali dan Monitoring melalui Serial USB

1. Rangkaian masih sama seperti sebelumnya di II.4.2.1

2. Buka program sebelumnya “ESP8266_InputDigital_buzzer” dari percobaan II.4.2.1. lalu Save as “ESP8266_InputDigital_buzzer_Serial”, edit sbb;

```
int baca, nilai,x; //buat variable baca sensor
String text="";
void setup(){
    Serial.begin(115200); //Aktifkan Serial di baudrate (bps)
    pinMode(D4, OUTPUT); //Set Pin D4 sbg Output buzzer
    pinMode(D1, INPUT_PULLUP); //Set Pin D1 sbg Input LDR
}
void loop() {
    baca=digitalRead(D1); //baca pin-D1 simpan di var baca
    if (baca==1){ //jika var baca logic 1 artinya ada bayangan
        digitalWrite(D4, LOW); //hidupkan buzzer
        Serial.println("ada bayangan"); //kirim test ke serial
    } //jika tidak logic 0 artinya gelap
    else{ //selain itu
        digitalWrite(D4, HIGH); //matikan buzzer
    }
    if(Serial.available()){ // jika ada serial masuk
        delay(10); text="";
        while(Serial.available()){
            text+=(char)Serial.read(); // baca isi serial
        }
        nilai=text.toInt(); //ubah dari text jadi angka
        Serial.print("setting buzzer: "); //kirim ke serial
        Serial.println(nilai);
        if (nilai>0){
            digitalWrite(D4,LOW); delay(nilai); //buzzer aktif
        }else{
            digitalWrite(D4,HIGH); delay(5000); //buzzer mati 5s
        }
    }
}
```

3. Testing dan Observasi:

Pada langkah ini seharusnya seperti percobaan sebelumnya namun logika dibalik;

- Buzzer bersuara jika sensor LDR gelap tertutup bayangan
- Buzzer tidak bersuara jika sensor LDR terang

Disertai fitur tambahan Monitoring dan Kendali melalui Serial USB;

- Buka Serial monitor, pada menu Tools->Serial monitor, atau Ctrl+Shift+M. Lalu atur baudrate sesuai program yakni 115200 bps.
- Melalui serial monitor kita bisa monitoring LDR, LDR tertutup bayangan akan terlihat text “ada bayangan”
- Melalui serial monitor kita bisa mengendalikan On/Off pada buzzer, jika buzzer sedang mati pada message box ketikan “1000” agar buzzer aktif selama 1000ms alias 1s. Silahkan coba angka lainnya. Jika sedang bunyi, kirim 0 untuk mematikan selama 5s.

4. Pengembangan kasus, buat serial untuk enable/disable sistem keamanan.

~~II.4.2.3 Kendali dan Monitoring melalui Wifi ESP sebagai Webserver Lokal~~

1. Rangkaian masih sama seperti sebelumnya di II.4.2.1

2. Buka program example dari menu

File->Examples->ESP8266WebServer->HelloServer

3. Ubah beberapa bagian program berikut:

```
#define STASSID "your-ssid" //SSID hotspot wifi yg tersedia
#define STAPSK "your-password" //password hotspot wifi-nya
const int led = 13; //menjadi D4 (lampu LED_BuiltIn)
```

4. Tambah delay pada bagian program berikut:

```
void handleRoot() {
    digitalWrite(led, 1); //off
    server.send(200, "text/plain", "hello from esp8266!");
    delay(5000);
    digitalWrite(led, 0); //on
}
```

5. Tambahkan mematikan led pada bagian program berikut:

```
server.on("/inline", []() {
    server.send(200, "text/plain", "this works as well");
    digitalWrite(led, 1); //off
});
```

6. Tambahkan menghidupkan led pada bagian program berikut:

```
server.on("/gif", []() {
    digitalWrite(led, 0); //on
    static const uint8_t gif[] PROGMEM = {
```

7. Upload Program.

8. Testing dan Observasi. Perhatikan dan pahami setiap baris programnya,

- LED Biru dekat antena akan ON.
- Buka Tools->Serial Monitor atau Ctrl+Shift+M

Karena pada void setup() menggunakan “Serial.begin(115200);”, maka pada serial Monitor pun harus disesuaikan ke baudrate 115200 bps (bits persecond).

Akan terlihat tampilan	Artinya
.....	menghubungkan ke wifi
Connected to yours-ssid	terhubung pada yours-ssid
IP address: 192.168.3.19	IP Web server
MDNS responder started	service MDNS on
HTTP server started	server sudah bisa di akses

*Jika tampilan diatas tidak muncul atau hanya muncul sebagian, tekan tombol RST (restart) pada board ESP8266.

*Lanjutkan pengujian di halaman selanjutnya

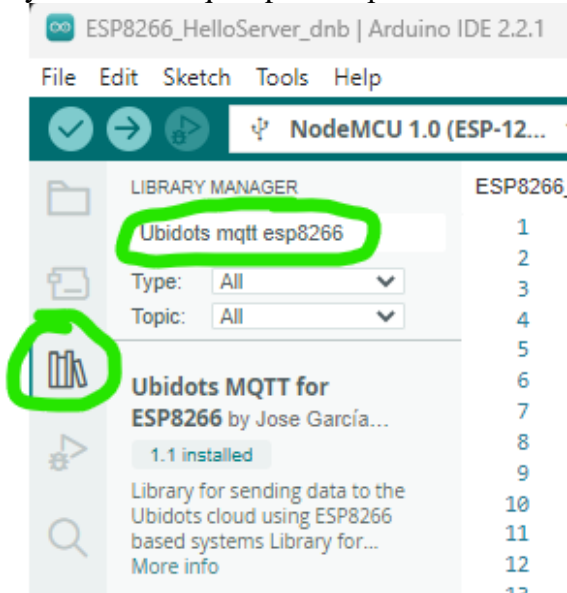
- Buka pada web browser smartphone alamat dengan IP webserver diatas;

Alamat diakses	Tampil pada browser	Led NodeMCU
192.168.3.19	hello from esp8266!	Off 5 detik, On
192.168.3.19/inline	this works as well	Off
192.168.3.19/gif	layar hitam dan emot	On
192.168.3.19/bebas	File Not Found	Tidak berubah

9. Pengembangan kasus;
Buatlah sublink baru /kedip membuat Led berkedip 3 kali

II.4.2.4 Kendali dan Monitoring melalui Ubidots IoT Platform

1. Rangkaian masih sama seperti sebelumnya di II.4.2.1, masih terhubung dengan buzzer dan LDR.
2. Kita install library “Ubidots mqtt esp8266” pada IDE Arduino



Gambar II.12 Install Library Ubidots MQTT for ESP8266

3. Buka program example dari menu:
File->Examples-> Ubidots MQTT for ESP8266->subscribe_and_publish
Save as “ESP8266_Ubidots_LDR_Buzzer”
4. Untuk menghubungkan NodeMCU ke Cloud ada 5 hal penting yang perlu di sesuaikan antara Alat kita dengan IoT Platform-nya, diantaranya sbb. :
 - 1) Nama SSID Wifi,
 - 2) Password Wifi
 - 3) Token IoT Platform
 - 4) Nama Device, umumnya hanya 1 saat latihan
 - 5) Nama Variabel sensor/aktuator, ini bisa lebih dari 1.

5. Sesuaikan dan ubah beberapa bagian program berikut yang di-highlight:

a). Pada bagian awal

```
#define TOKEN "BBUS-XngSGdvsk8AGhxxx" //Your Ubidots TOKEN
#define WIFINAME "Sejuk" //Your SSID
#define WIFIPASS "lupalagi" // Your Wifi Pass
const int led=D4; // penulisan D4 menjadi led
```

b). Pada bagian Auxiliar Functions

```
//***** Di bagian Auxiliar Functions *****
void callback(char* topic, byte* payload, ....
    digitalWrite(led, HIGH); //16 diganti jadi led
    digitalWrite(led, LOW); //16 diganti jadi led
```

c). Pada bagian setup()

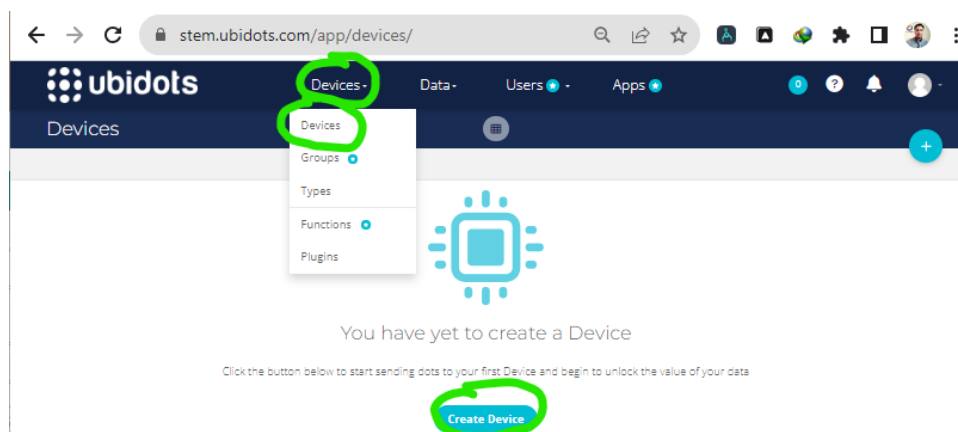
```
//***** Di bagian setup() *****
void setup() {...
    pinMode(led, OUTPUT); //16 diganti jadi led
    pinMode(D1, INPUT_PULLUP); //Set D1 sebagai input (LDR)
    client.ubidotsSubscribe("esp8266_dnb", "buzzer");
    //(nama device label, nama variable label)
```

d). Pada bagian loop()

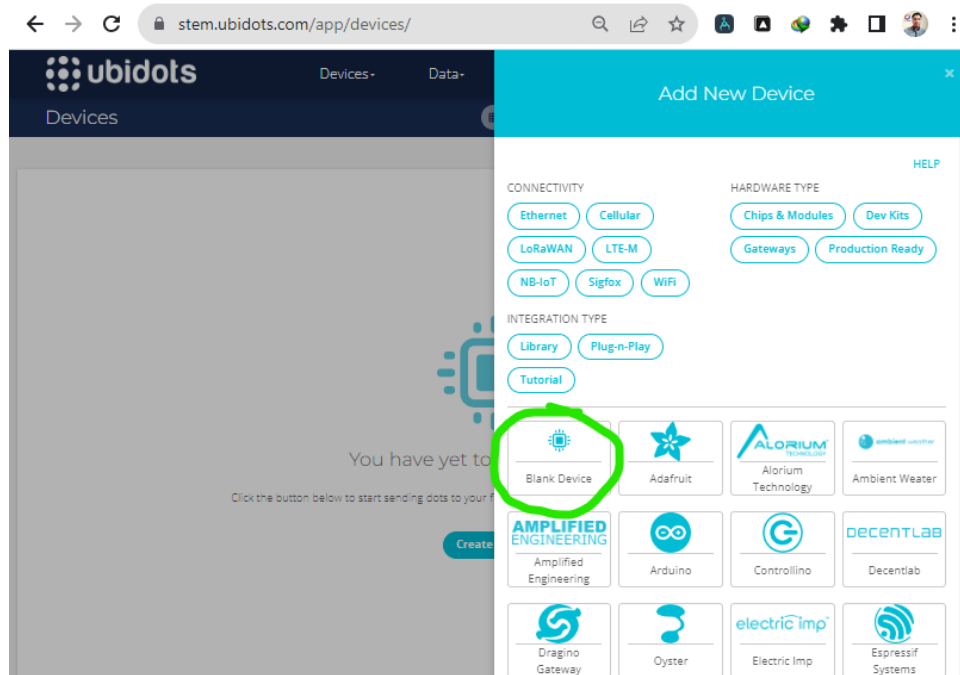
```
void loop() {
    if(!client.connected()){
        client.reconnect();
        client.ubidotsSubscribe("esp8266_dnb", "buzzer");
        //(nama device label, nama variable label)
    }
    client.add("ldr", digitalRead(D1)); //data ldr dari D1
    //max send 4.000 dots/d (Monitoring)
    client.ubidotsPublish("esp8266_dnb");
    client.loop(); //max receive 500.000 dots/d (Control)
    delay(5000); //hemat kuota, 21s=26jam, 10s=12jam, 5s=6jam
}
```

6. Buat aplikasi di bagian Cloud, IoT webserver yang kita gunakan Ubidots.

1) Buat Device baru pada akun Ubidots Stem



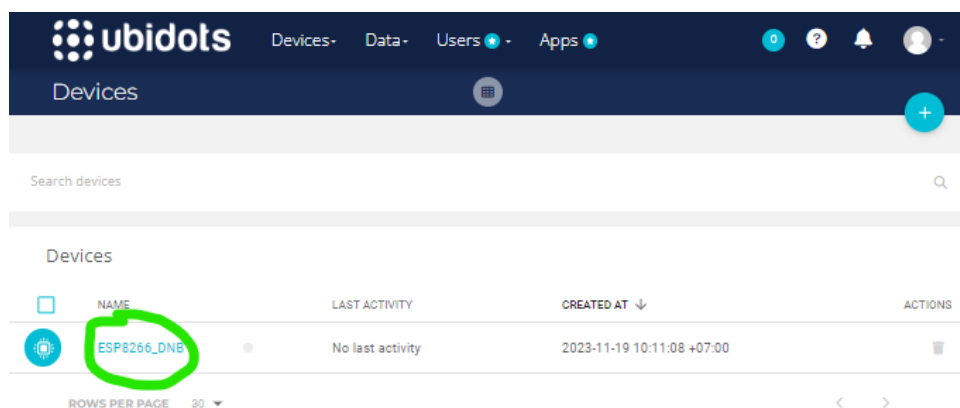
2) Pilih Blank Device



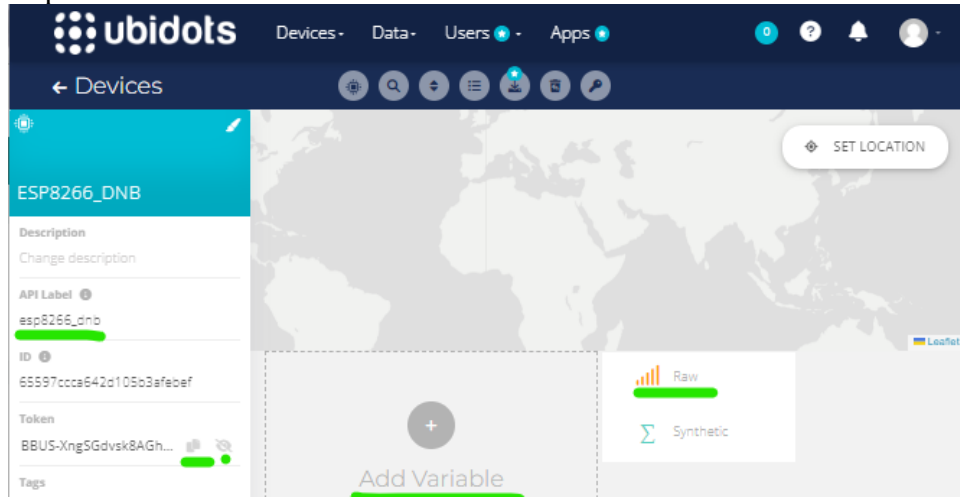
3) Isikan nama device, dan device label akan mengikuti dengan huruf kecil

The screenshot shows the 'Add New Device' form. The 'Device name' field is filled with 'ESP8266_DNB' and the 'Device label' field is filled with 'esp8266_dnb'. A green checkmark icon is visible at the bottom right.

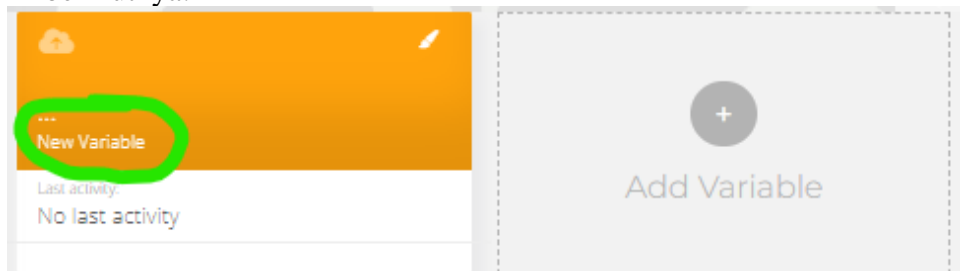
4) Klik device yang sudah dibuat.



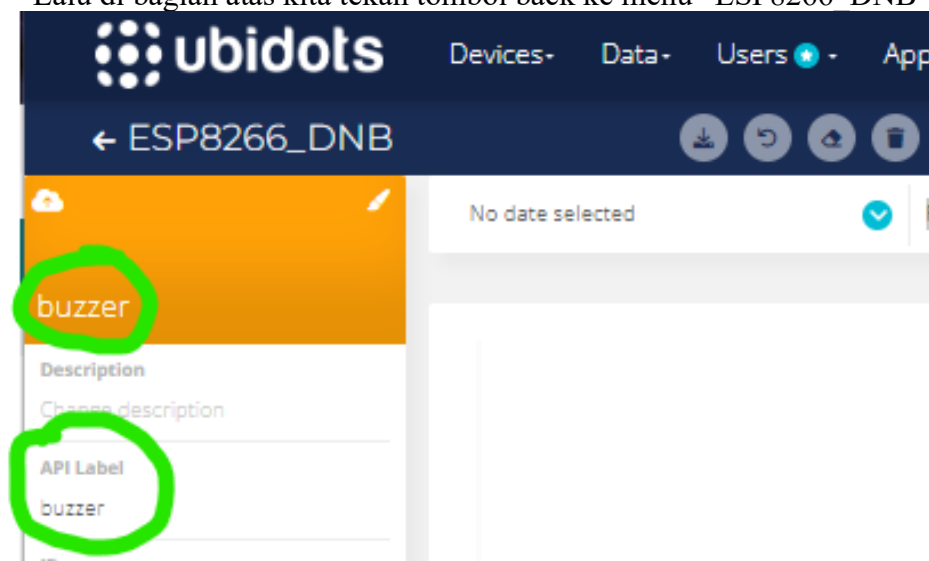
- 5) Perlu diingat nama **device label**-nya, misalnya pada contoh ini “esp8266_dnb” dan tokennya, **token** bisa dilihat dengan menekan tombol mata dan bisa dicopy dengan menekan tombol copy. Selanjutnya tambahkan variable baru dengan click “add variable”, lalu pilih “Raw”



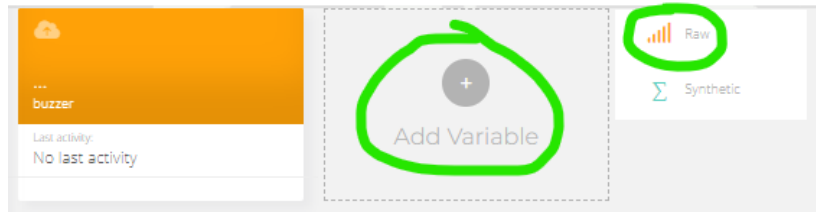
- 6) Selanjutnya akan muncul variable baru, bisa kita ubah dengan cara double click pada “New Variable” menjadi nama variable yang kita inginkan, atau sekali click untuk setting lebih lanjut, lihat gambar berikutnya.



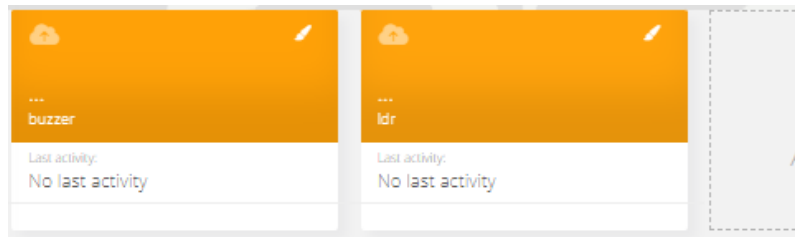
- 7) Pada New Variable tadi setelah di click 1 kali, selanjutnya kita setting, namun kali ini hanya kita ganti saja sesuai desain kita, sisanya default. Lalu di bagian atas kita tekan tombol back ke menu “ESP8266 DNB”



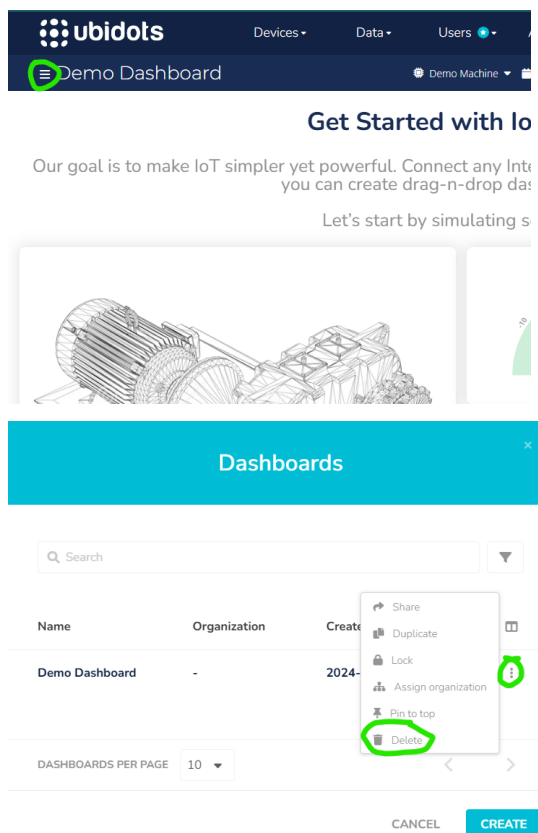
- 8) Disini kita bisa menambahkan variable baru lagi dengan cara yang sama, click “Add Variable” lalu pilih “Raw”



- 9) Kita rename “New Variable” tersebut menjadi LDR dalam huruf kecil menjadi “ldr”



- 10) Lalu kita ke halaman Dashboard pada menu Data-> Dashboards, **Jika Sudah Ada Dashboard Sebelumnya** silahkan hapus dahulu, click tombol 3 garis di atas kiri atas, lalu click tombol 3 titik dan click delete.

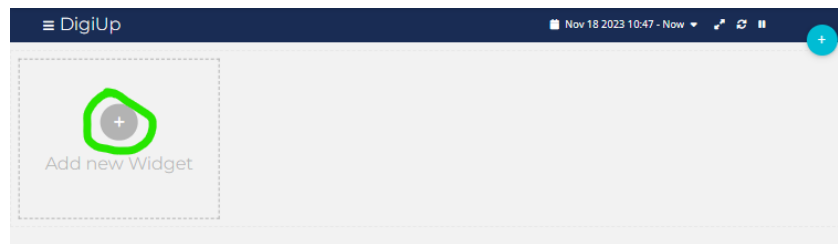


- 11) Lalu kita buat dashboard baru, click “Add new Dashboard”. Kita beri nama Dashboard ini “DigiUp”, lalu tekan tombol save

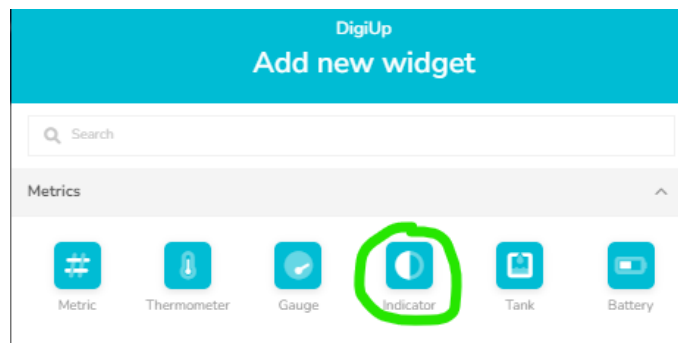
The first screenshot shows the Ubidots interface with the 'Data' menu open and 'Dashboards' selected. The second screenshot shows the 'Add new Dashboard' form with the following details:

SETTINGS	APPEARANCE
Name	DigiUp
tags custom identifiers that can be attached to one or more users.	tag1, tag2, tag3...
Default time range	Last 24 hours
Dynamic Dashboard Update widgets based on selected Device	Static
Width	Auto
Alignment	Center
Date format	11/19/2023 10:43
Filters	
+ ADD FILTER	
CANCEL SAVE	

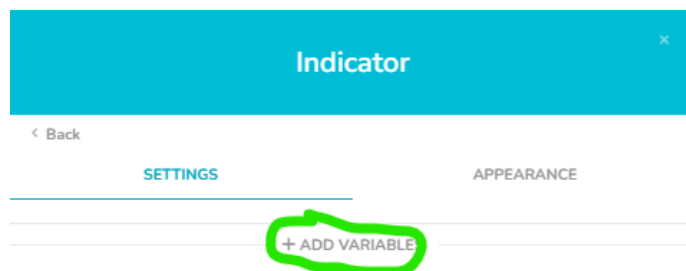
- 12) Selanjutnya dashboard baru ini sudah terbuka perlu ditambahkan widget LDR untuk kita monitor. Click “Add New Widget”



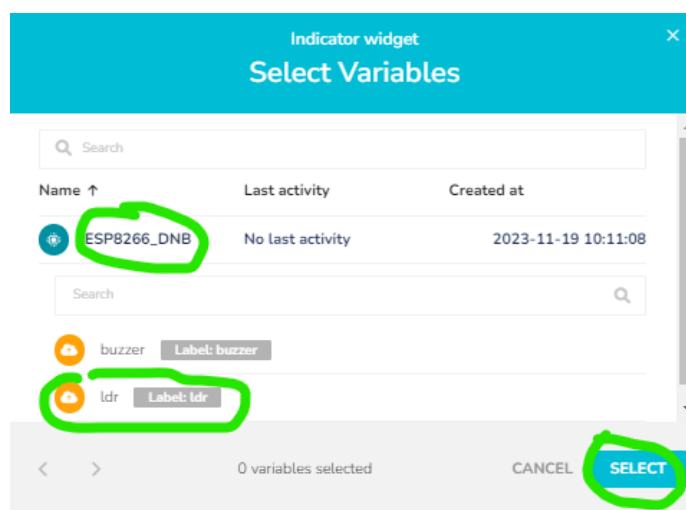
- 13) Lalu pilih “Indikator”,



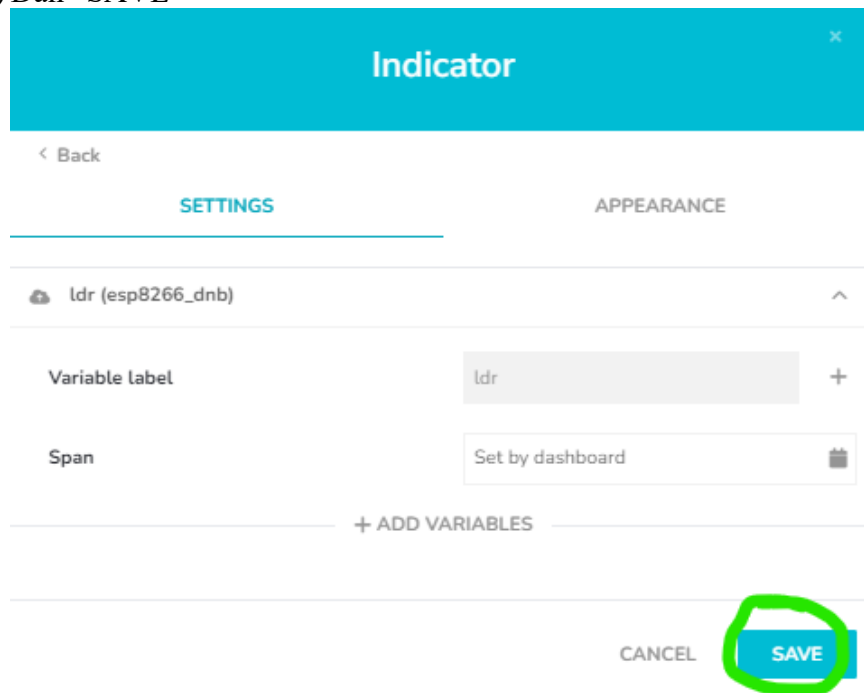
- 14) Lalu pilih “ADD VARIABLE”,



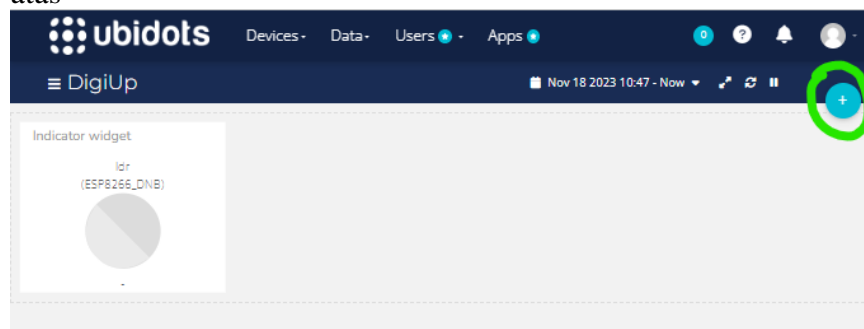
- 15) Selanjutnya click nama device-nya (ESP8266_DNB) dan pilih variable label yang diinginkan “ldr”, lalu tekan tombol “SELECT”



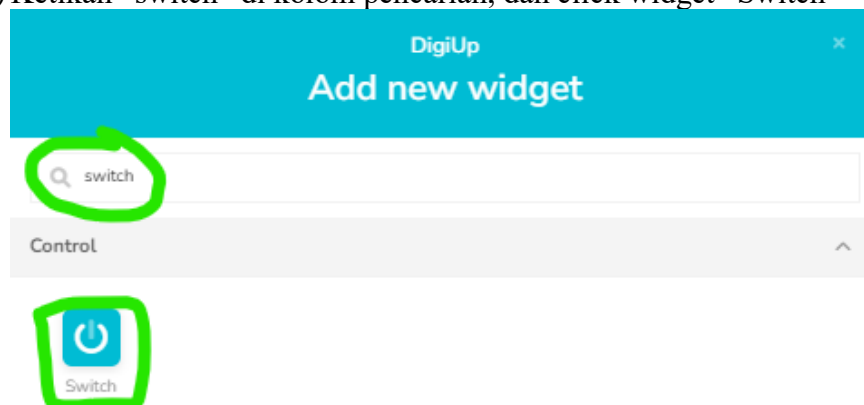
16) Dan “SAVE”



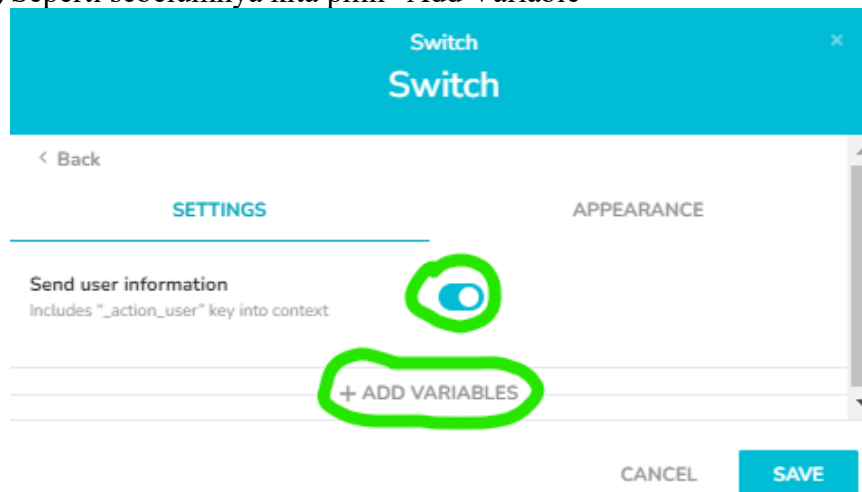
17) Selanjutnya kita melihat halaman dashboard lagi, dan kita tambahkan widget baru lagi untuk mengendalikan buzzer, click tombol “+” di kanan atas



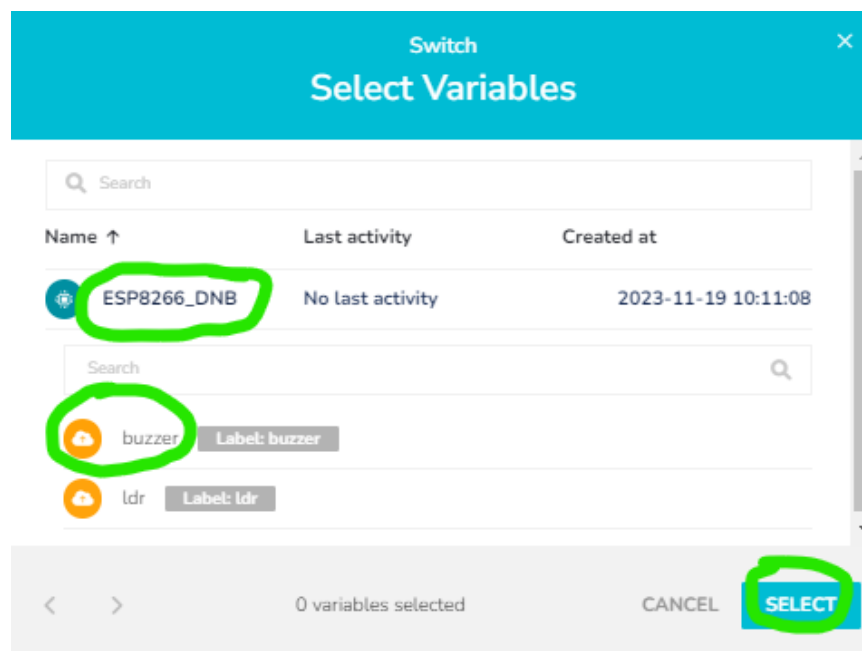
18) Ketikan “switch” di kolom pencarian, dan click widget “Switch”



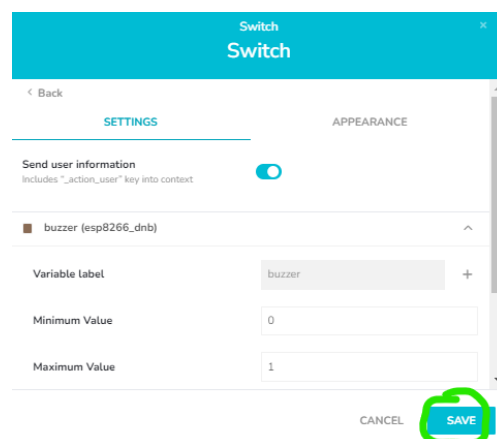
19) Seperti sebelumnya kita pilih “Add Variable”



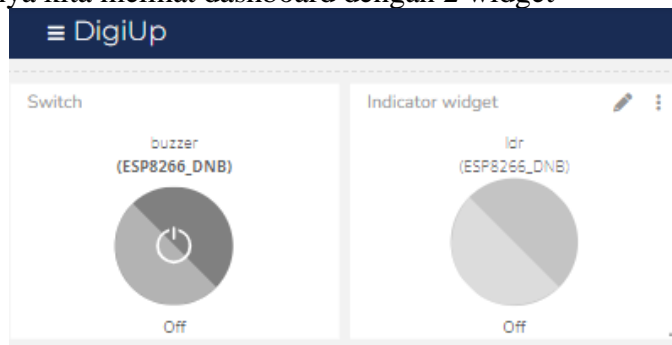
20) Ambil dari device ESP8266_DNB, lalu pilih “buzzer”, “SELECT”.



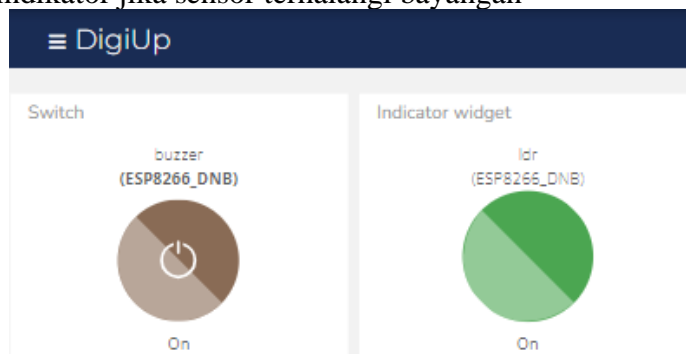
21) Lalu tekan SAVE



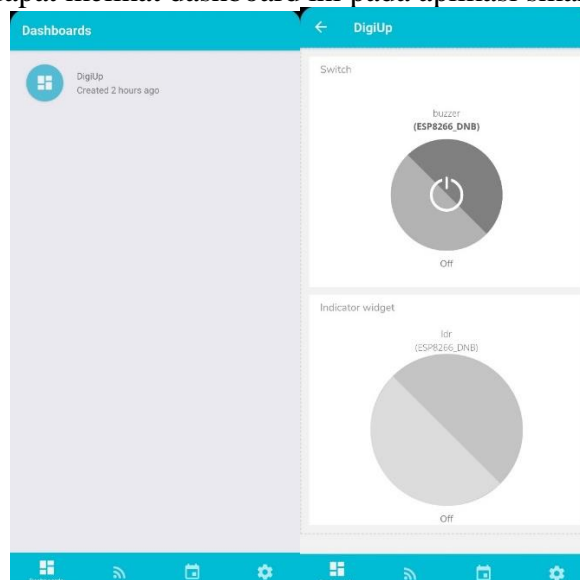
22) Selanjutnya kita melihat dashboard dengan 2 widget



23) Widget buzzer dapat kita click untuk mengendalikan buzzer, dan ldr sebagai indikator jika sensor terhalangi bayangan



7. Selanjutnya kita dapat melihat dashboard ini pada aplikasi smartphone.



10. Testing dan Observasi. Perhatikan dan pahami setiap bagian programnya,
- Perhatikan 3 bagian ini; Serial monitor, Website dan Aplikasi Ubidots
 - Pengendalian dan monitoring terjeda 5 detik.
 - Kita bisa **kendalikan** lampu Led/buzzer melalui Web browser dan HP
 - Kita bisa **monitoring** sensor cahaya LDR melalui Web browser dan HP

11. Kembangkan kasus agar bisa Sistem IoT bisa dikendalikan enable/disable melalui Serial.

II.5 Diskusi

Part III. Input Analog

III.1 Tujuan Workshop

1. Peserta mampu melakukan pemrograman dengan menggunakan logika percabangan dan pengulangan.
2. Peserta mampu mengolah dan menseleksi data.
3. Peserta mampu merakit potensiometer sebagai sensor analog dan relay sebagai aktuator pada sistem.
4. Peserta mampu mengeluarkan output PWM seolah-olah analog dari kontoler.

III.2 Dasar Teori

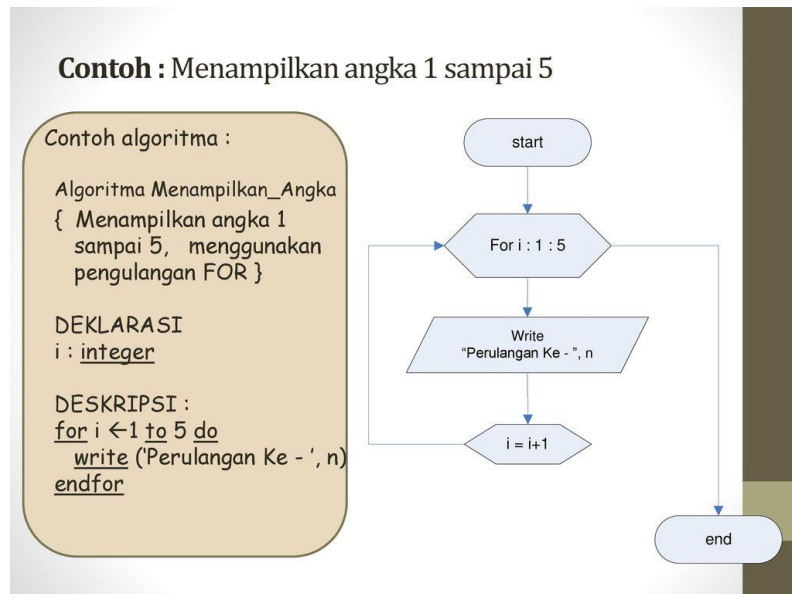
Algoritma adalah langkah-langkah logis untuk mencapai tujuan. Algoritma pemrograman dasar terdiri dari; Sekuensial, Percabangan dan Pengulangan. **Algoritma percabangan** digunakan untuk mengolah data dalam hal seleksi suatu data/program terhadap kondisi tertentu. Contoh Algoritma percabangan digambarkan pada gambar III.1.

Contoh Percabangan 2 Kasus (IF-THEN-ELSE)



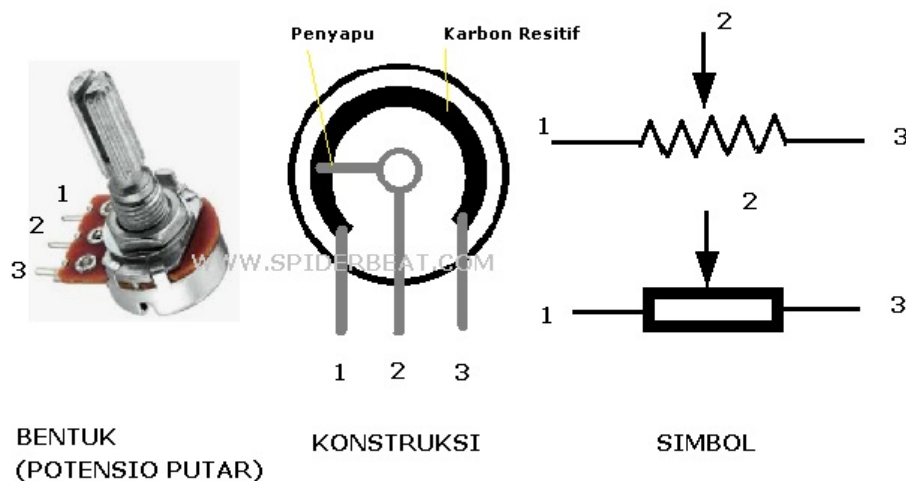
Gambar III.1. Contoh algoritma percabangan.

Algoritma perulangan digunakan untuk mengolah data dalam hal mengulang data/program terhadap kondisi tertentu. Contoh Algoritma perulangan digambarkan pada gambar III.2



Gambar III.2. Contoh algoritma perulangan.

Potensiometer adalah komponen elektronik yang dapat mengubah resistansinya sesuai dengan perubahan posisi pengaturannya. Sebagai sensor analog, potensiometer dapat digunakan untuk mengukur perubahan posisi atau keadaan fisik tertentu, seperti rotasi atau pergeseran. Prinsip kerjanya berdasarkan pada pembagian tegangan atau arus dalam suatu rangkaian resistansi variabel.



Gambar III.3 Potensiometer

Pembacaan Analog pada NodeMCU sebenarnya menggunakan fitur Analog to Digital Converter (ADC). ADC yang digunakan 10 bit sehingga $2^{10}=1024$ kombinasi, yakni 0 sd 1024 sehingga nilai minimum sd maksimum akan dikonversi

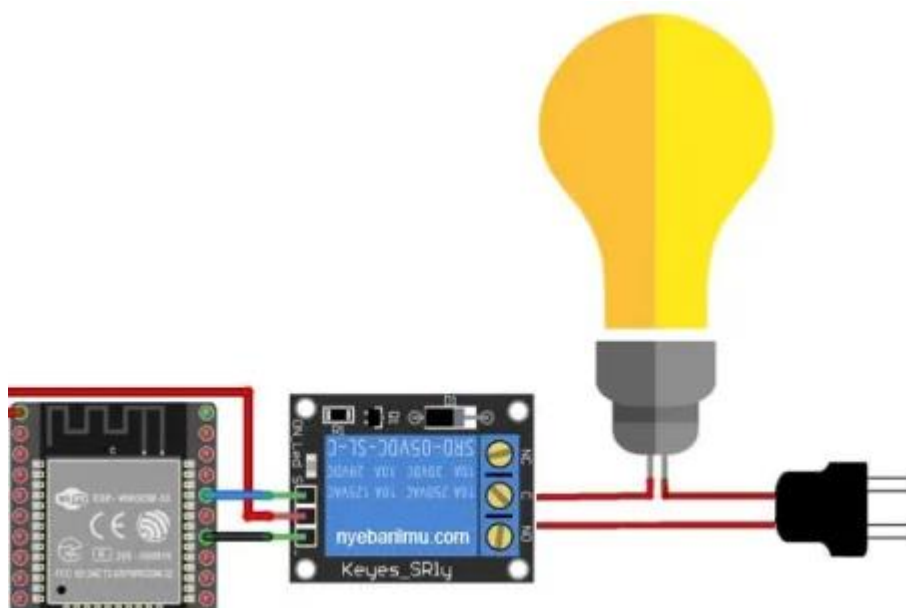
menjadi nilai tersebut. Karena NodeMCU beroperasi menggunakan 0 sd 3.3 Volt maka;

- Nilai ADC 0 adalah 0 volt
- Nilai ADC 1024 adalah 3.3 volt

Relay adalah perangkat elektromekanis yang bekerja sebagai saklar yang dioperasikan secara listrik. Relay dapat digunakan sebagai aktuator untuk mengendalikan perangkat atau sistem lain berdasarkan sinyal kontrol listrik. Prinsip kerja relay melibatkan elektromagnet dan kontak saklar yang dapat membuka atau menutup tergantung pada aktivasi elektromagnetiknya.

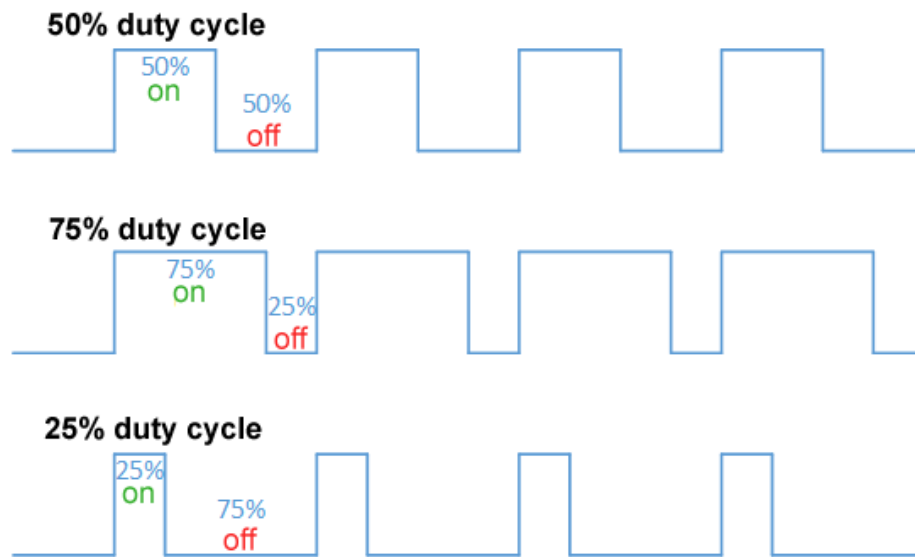


Gambar III.4 Relay 1 Channel



Gambar III.5 Rangkaian Relay 1 Channel

PWM (Pulse Width Modulation) adalah metode pengkodean sinyal di mana lebar pulsa sinyal dibuat bervariasi, sementara frekuensinya tetap konstan. Output PWM dapat dianggap "seolah-olah analog" karena lebar pulsa yang berubah-ubah dapat mensimulasikan sinyal analog dengan mengendalikan rasio antara waktu sinyal aktif dan waktu tidak aktifnya. PWM sering digunakan untuk mengendalikan kecepatan motor, intensitas cahaya, atau sistem kontrol lainnya.



Gambar III.6 Pulse Width Modulation

III.3 Alat dan Komponen yang Digunakan

- 1) Laptop/PC dengan Software IDE Arduino Lastest
- 2) NodeMCU + USB Cable 1 buah
- 3) Potensiometer 1 buah
- 4) Kabel Jumper Female-Female 6 buah
- 5) Modul Relay 1 buah

III.4 Langkah-Langkah Percobaan

III.4.1 Langkah Persiapan

1. Tersedia IDE Arduino, pastikan sudah terinstal sudah terinstal ESP8266 board, pilih dengan cara:
Tools->Board->ESP8266 board->NodeMCU1.0(ESP-12E Module)

2. Tersedia Laptop/PC, pastikan hardware ESP8266 sudah terinstal. Buka “Device Manager” pada windows anda, dan lihat nomor COM yang terhubung.

III.4.2 Langkah Workshop

III.4.2.1 Membaca Input Analog

1. Rangkailah potensiometer dan relay pada NodeMCU seperti pada tabel berikut:

ESP8266	Potensiometer
G	1
A0	2
3V	3

ESP8266	Relay
G	-
3V	+
D4	S

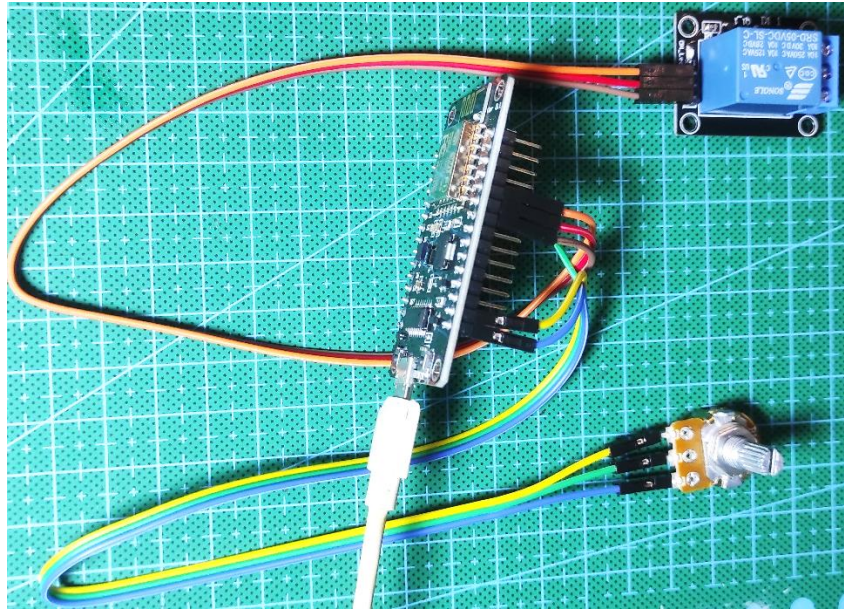


BENTUK
(POTENSIO PUTAR)



KONSTRUKSI

Hati-hati Pin No.2 (tengah) tidak boleh terhubung ke Vin, 3V atau Ground, berisiko merusak Laptop (Short)



2. Buat file baru pada IDE Arduino, dan simpan dengan nama ESP8266_InputAnalog
3. Ketik dan Upload code program berikut:

```
int analog = 0;
void setup() {
  Serial.begin(115200);
}
void loop() {
  analog = analogRead(A0); //baca analog dari Pin A0
  Serial.println(analog); //kirim data analog ke Serial
  delay(100);
}
```

4. Testing dan Observasi
 - Buka Tools->Serial Monitor atau Ctrl+Shift+M
 - Pada langkah ini seharusnya, data tampil data ADC 0 sd 1024, saat potensiometer diputar CCW maka tegangannya mengecil sehingga data ADC yang tampil juga akan mengecil, sedangkan saat potensiometer diputar CW maka tegangannya membesar sehingga data ADC yang tampil juga akan membesar.
 - Jika berhasil coba lihat juga pada serial plotter, dan lakukan seperti pada poin sebelumnya. Tools->Serial Plotter.

III.4.2.2 Mengolah data Input Analog ke Relay

1. Rangkaian masih seperti pada percobaan sebelumnya.
2. Buat file baru pada IDE Arduino, dan simpan dengan nama ESP8266_InputAnalog_Seleksi

3. Ini adalah program sebelumnya yang di kembangkan, Ketik dan Upload code program berikut:

```
int analog = 0;
void setup() {
  Serial.begin(115200);
  pinMode(D4, OUTPUT);
}
void loop() {
  analog = analogRead(A0); //baca analog dari Pin A0
  Serial.println(analog); //kirim data analog ke Serial
  if (analog>500) //Jika analog diatas 500
    digitalWrite(D4, 1); //Relay On
  else //Jika tidak
    digitalWrite(D4, 0); //Relay Off
  delay(100);
}
```

4. Testing dan Observasi
 - Buka Tools->Serial Monitor atau Ctrl+Shift+M
 - Serial plotter, dan lakukan seperti pada percobaan sebelumnya memutar kira dan kanan pada potensio meter.
 - Jika data Analog diatas 500 maka relay akan On, jika dibawah 500 Relay Off.
5. Pengembangan kasus, jika analog kurang dari 300 mati, antara 300 sd 600 relay berkedip, diatas 600 relay On

III.4.2.3 Monitoring Analog dan Kendali Digital Melalui Ubidots

1. Rangkaian masih seperti pada percobaan sebelumnya.
2. Buka program sebelumnya ESP8266_Ubidots_LDR_Buzzer dari percobaan II.4.2.4
Save as “ESP8266_Ubidots_Potensiometer_Relay”
3. Untuk menghubungkan NodeMCU ke Cloud ada 5 hal penting yang perlu di sesuaikan antara Alat kita dengan IoT Platform-nya, diantaranya sbb. :
 - a. Nama SSID Wifi,
 - b. Password Wifi
 - c. Token IoT Platform
 - d. Nama Device, umumnya hanya 1 saat latihan

e. Nama Variabel sensor/aktuator, ini bisa lebih dari 1.

Sesuaikan, ubah/tambahkan beberapa bagian programnya yang di highlight:

```
#define TOKEN "BBUS-XngSGdvsk8AGhgrm0"//Your Ubidots TOKEN
#define WIFINAME "Sejuk" //Your SSID
#define WIFIPASS "lupalagi" // Your Wifi Pass
const int led=D4; // penulisan D4 menjadi led
int analog;

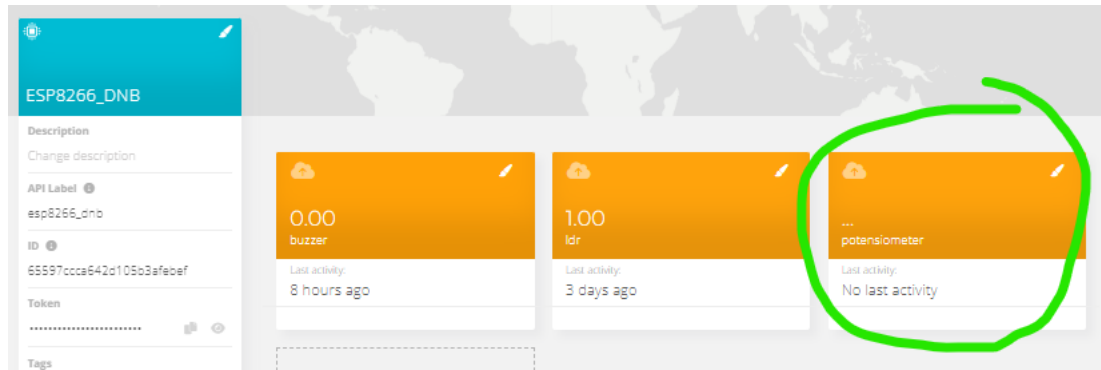
//***** Di bagian Auxiliar Functions *****
void callback(char* topic, byte* payload, ....
    digitalWrite(led, HIGH); //16 diganti jadi led
    digitalWrite(led, LOW); //16 diganti jadi led

//***** Di bagian Main Function *****
void setup() {...
    pinMode(led, OUTPUT); //16 diganti jadi led
    //pinMode(D1,INPUT_PULLUP); // hapus tidak digunakan
    client.ubidotsSubscribe("esp8266_dnb","buzzer"); //(nama
    device label, nama variable label)

void loop() {
    if(!client.connected()){
        client.reconnect();
        client.ubidotsSubscribe("esp8266_dnb","buzzer"); //(nama
    device label, nama variable label)
    }
    analog = analogRead(A0); //baca analog dari Pin A0
    Serial.println(analog); //kirim data analog ke Serial
    client.add("potensiometer", analog); //data potensiometer
    dari A0
    client.ubidotsPublish("esp8266_dnb"); //max send 4.000
    dots/d (Monitoring)
    client.loop(); //max receive 500.000 dots/d (Control)
    delay(5000); //hemat kuota, 21s=26jam, 10s=12jam, 5s=6jam
    }
```

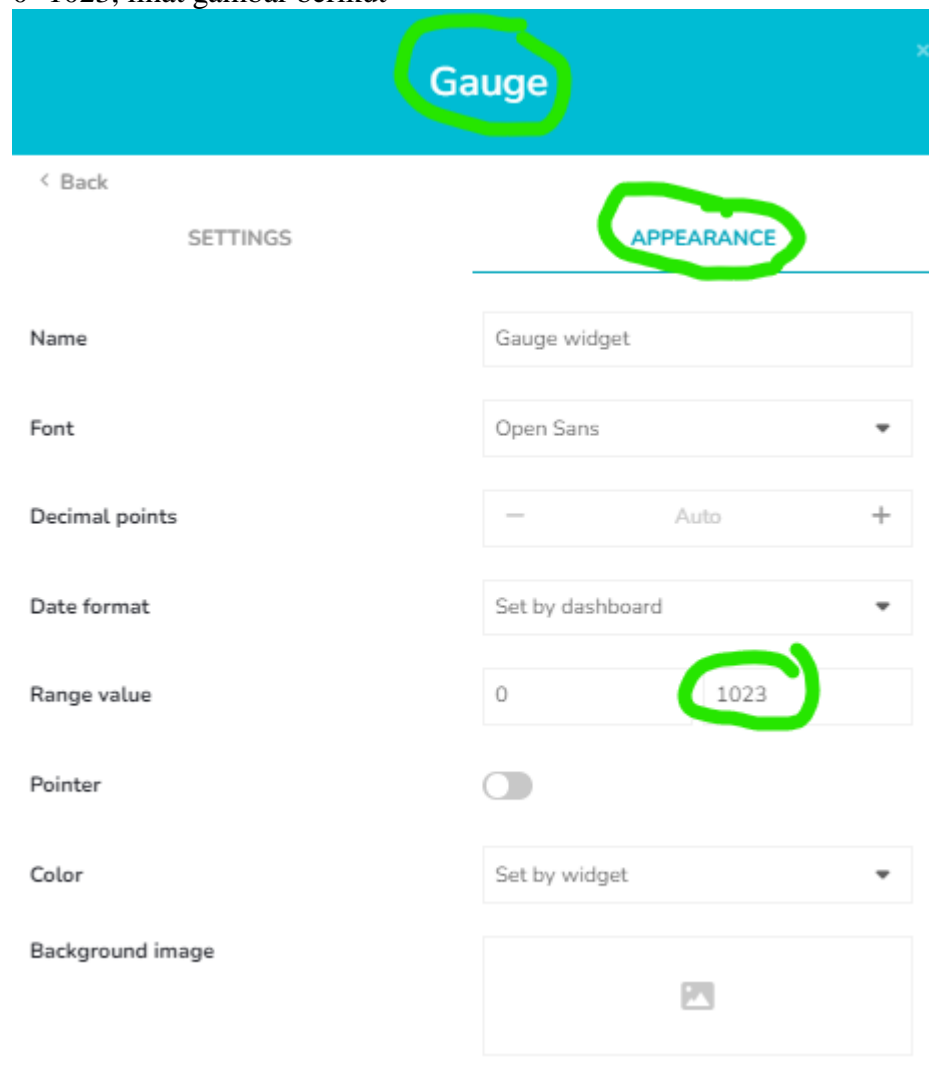
4. Buat aplikasi di bagian Cloud, IoT webserver yang kita gunakan Ubidots.

- 1) Device pada akun Ubidots Stem masih menggunakan yang sebelumnya di Part II “esp8266_dnb.
- 2) Masih menggunakan token sebelumnya
- 3) Pada menu device:
 - Variable sebelumnya “led” bekas buzzer bisa digunakan untuk Relay karena sama-sama output digital
 - Namun kita perlu menambahkan variable baru “potensiometer” untuk potensiometer, dengan cara yang sama pada part II



4) Pada menu dashboard

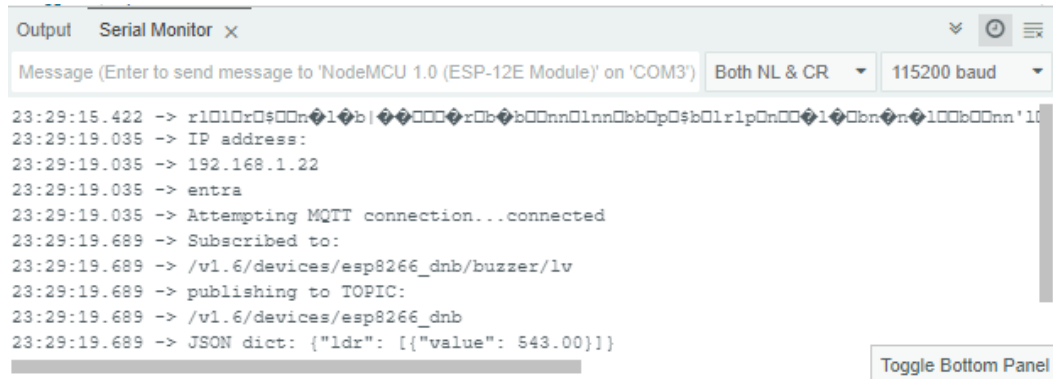
- kita tambahkan widget baru yaitu “Gauge” dengan add variable “potensiometer”, caranya bisa lihat part II. Sebelum tekan “save” lakukan di bawah ini
- Gauge ini kita atur pada bagian appearancenya, agar range value 0 -1023, lihat gambar berikut



5. Testing dan Observasi

- Buka Tools->Serial Monitor atau Ctrl+Shift+M

- Serial plotter, cek apakah berhasil terhubung ke wifi, apakah terhubung ke ubidots. Berikut contoh tampilan jika berhasil terhubung ke wifi dan ubidots



```
23:29:15.422 -> r10l0r0$00n0l0b|0000r0b0b00nn0lnn0bb0p0$b0lrlp0n000l00bn0n0l00b00nn'10
23:29:19.035 -> IP address:
23:29:19.035 -> 192.168.1.22
23:29:19.035 -> entra
23:29:19.035 -> Attempting MQTT connection...connected
23:29:19.689 -> Subscribed to:
23:29:19.689 -> /v1.6/devices/esp8266_dnb/buzzer/lv
23:29:19.689 -> publishing to TOPIC:
23:29:19.689 -> /v1.6/devices/esp8266_dnb
23:29:19.689 -> JSON dict: {"ldr": [{"value": 543.00}]}
```

- Serial plotter, dan lakukan seperti pada percobaan sebelumnya memutar kira dan kanan pada potensiometer. Akan nampak data baru setiap 5 detik, begitupun tampilan ubidots akan berubah sesuai inputan.
 - Saat potensiometer dioperasikan perhatikan pada Web Ubidots dan layar Smartphone. Data akan update setiap 5 detik sesuai program kita
 - Kendalikan On/Off Relay melalui tombol buzzer di ubidots. Relay akan merespon setiap 5 detik sesuai program kita.
6. Pengembangan, silahkan program arduinonya save as “..._pengembangan” dan pada program ini sesuaikan nama led dan buzzer menjadi relay, ubidotsnya.

Tunjukan hasil perubahannya, lalu uji kembali seperti testing dan observasi tugas sebelumnya.

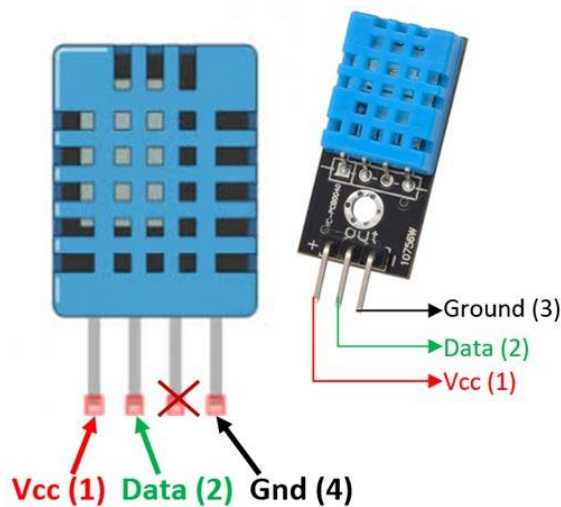
Part IV. Sensor Jarak dan Output Analog

IV.1 Tujuan Workshop

1. Peserta mampu mengeluarkan output PWM seolah-olah analog dari kontoler.
2. Peserta mampu menggunakan DHT11 sebagai sensor suhu dan kelembaban
3. Peserta mampu menggunakan sensor ultrasonik sebagai sensor jarak
4. Peserta mampu menggunakan servo sebagai pengaturan gerak sudut
5. Peserta mampu menyesuaikan tampilan pada Smartphone untuk monitoring dan kendali IoT.

IV.2 Dasar Teori

DHT11 adalah sensor suhu dan kelembaban yang sering digunakan dalam berbagai proyek elektronika dan IoT.



Gambar IV.1 DHT11

Prinsip Kerja DHT11 menggunakan termistor sebagai sensor suhu juga menggunakan higrometer untuk mengukur kelembaban. Termistor adalah sebuah resistor yang nilai resistansinya berubah secara signifikan dengan perubahan suhu. DHT11 memiliki termistor yang mengubah resistansinya sesuai dengan suhu sekitar. Higrometer adalah sensor yang mengukur kadar air atau kelembaban di

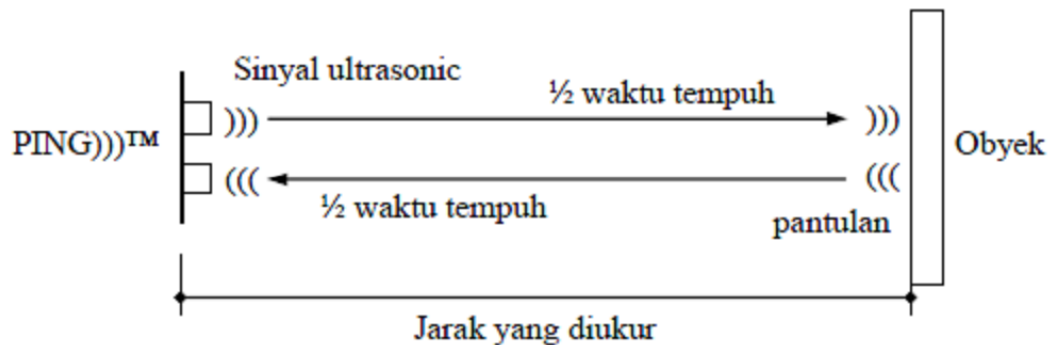
udara. DHT11 menggunakan higrometer resistif yang juga mengubah resistansinya sesuai dengan kelembaban udara.

DHT11 menggunakan satu kabel data untuk berkomunikasi dengan mikrokontroler atau perangkat lainnya. Protokol komunikasi yang digunakan adalah protokol satu kawat (single-wire) yang bersifat berseri (serial). Data dikirimkan dalam bentuk pulsa yang dipancarkan oleh sensor. Setiap pulsa mewakili bit data. Setiap pembacaan dari DHT11 menghasilkan data berupa 40 bit. Data ini terdiri dari suhu (16 bit), kelembaban (16 bit), dan checksum (8 bit). Format data yang dikirim adalah MSB (Most Significant Bit) dulu, dan setelahnya adalah LSB (Least Significant Bit). DHT11 dapat memberikan pembacaan suhu dan kelembaban pada interval tertentu. Interval pembacaan dapat bervariasi tergantung pada spesifikasi penggunaan atau program yang dijalankan. DHT11 memiliki presisi tertentu dalam pengukuran suhu dan kelembaban, dan batas pengukuran yang tergantung pada model dan spesifikasi tertentu. Kelebihan DHT11 meliputi harga yang terjangkau, ukuran kecil, dan kemudahan penggunaan. Namun, keterbatasan termasuk presisi yang mungkin tidak sesuai untuk aplikasi yang membutuhkan akurasi tinggi.

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran fisis (bunyi) menjadi besaran listrik. Rangkaian sensor ultrasonic terdiri dari Transmitter, Receiver, dan komparator. Ultrasonic modul umumnya berbentuk papan elektronik ukuran kecil dengan beberapa rangkaian elektronik dan 2 buah transducer. Dari 2 buah transducer ini, salah satu berfungsi sebagai transmitter dan satu lagi sebagai receiver. Ada juga modul yang hanya mempunyai 1 buah transducer, berfungsi sebagai transmitter dan receiver sekaligus. Ultrasonic modul ini bekerja dengan cara menghasilkan gelombang suara pada frekuensi tinggi, yang kemudian dipancarkan oleh bagian transmitter.

Pantulan gelombang suara yang mengenai benda di depannya akan ditangkap oleh bagian receiver. Dengan mengetahui lamanya waktu antara dipancarkannya gelombang suara sampai ditangkap kembali, kita dapat menghitung jarak benda yang ada di depan modul tersebut. Lamanya waktu tempuh gelombang suara dikalikan kecepatan suara, kemudian dibagi 2 akan menghasilkan jarak antara

ultrasonic modul dengan benda di depannya. Sensor Ultrasonic module HC-SR04 adalah suatu device atau modul yang berfungsi untuk mengukur jarak dengan cara memancarkan sinyal ultrasonik. Dengan dimensi yang cukup kecil yaitu 43x30x15 mm, serta harga yang sangat terjangkau. Modul HC-SR04 ultrasonic memiliki range atau mengukur jarak 2cm - 400 cm, kisaran akurasi mencapai 3mm.



Gambar IV.2 Ilustrasi Perhitungan Sensor Ultrasonik

Cara menghitung jarak pada sensor ultrasonik:

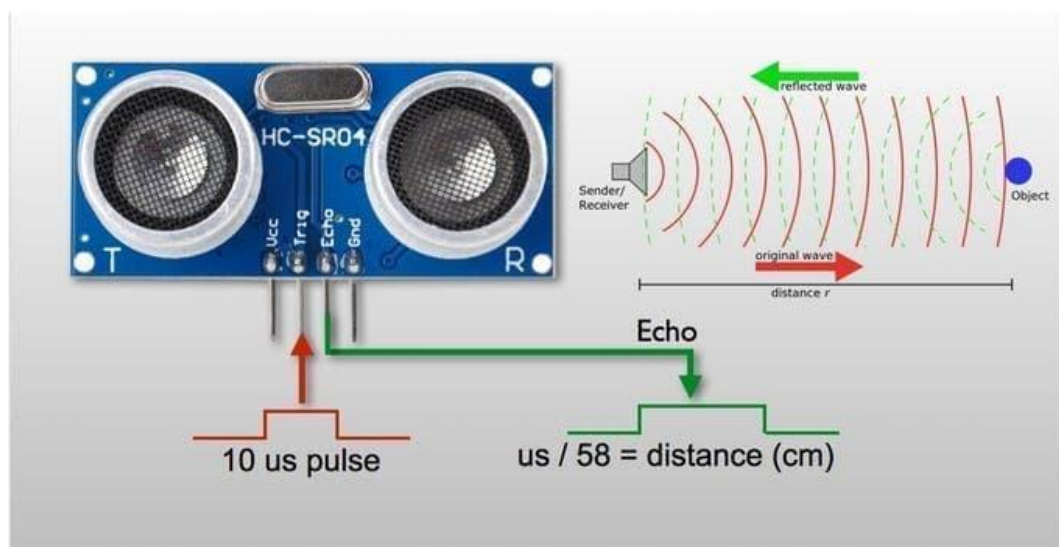
$$S = (tIN \times V) / 2$$

Dimana :

S = Jarak antara sensor ultrasonik dengan objek yang dideteksi

V = Cepat rambat gelombang ultrasonik di udara (344 m/s)

tIN = Selisih waktu pemancaran dan penerimaan pantulan gelombang



Gambar IV.3 Sensor Ultrasonic

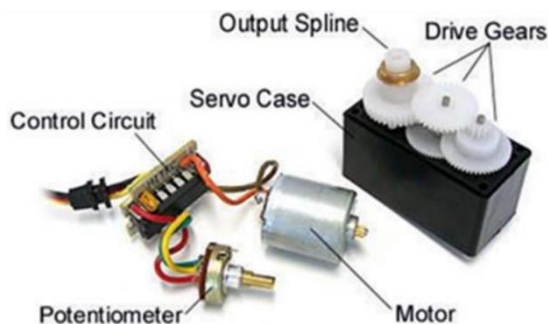
Sensor Ultrasonik sebagai Sensor Jarak bekerja berdasarkan prinsip gelombang ultrasonik yang dipancarkan dan dipantulkan untuk mengukur jarak antara sensor dan objek. Beberapa konsep dasar yang perlu dipahami dalam penggunaan sensor ultrasonik melibatkan:

1. Gelombang Ultrasonik, Sensor ultrasonik menggunakan gelombang suara dengan frekuensi tinggi, di luar kisaran pendengaran manusia (biasanya di atas 20 kHz). Gelombang ini dipancarkan ke objek, dan waktu yang dibutuhkan untuk gelombang tersebut kembali ke sensor digunakan untuk menghitung jarak.
2. Prinsip Time of Flight, Pengukuran jarak pada sensor ultrasonik didasarkan pada waktu yang diperlukan oleh gelombang ultrasonik untuk pergi ke objek dan kembali ke sensor. Jarak dapat dihitung dengan menggunakan rumus dasar waktu = jarak / kecepatan gelombang suara.
3. Ketidakpastian dan Pengaruh Lingkungan: Faktor-faktor seperti suhu, kelembaban, dan material objek dapat mempengaruhi kecepatan gelombang ultrasonik, sehingga perlu diperhatikan untuk mendapatkan hasil yang akurat.

Servo adalah perangkat mekanis atau elektromekanis yang digunakan untuk mengontrol gerakan sudut atau posisi suatu objek. Beberapa konsep dasar yang terkait dengan penggunaan servo untuk pengaturan gerak sudut melibatkan:

1. Pengaturan Sudut: Servo memiliki kemampuan untuk mengatur sudut atau posisi dengan presisi. Hal ini dicapai melalui pengaturan sinyal pulsa yang dikirimkan ke servo. Panjang pulsa ini menentukan posisi sudut tertentu.
2. Sinyal PWM (Pulse Width Modulation): Servo menerima sinyal PWM sebagai input kontrol. Panjang pulsa pada sinyal PWM menentukan posisi atau sudut yang diinginkan. Umumnya, pulsa dengan panjang 1 ms mewakili sudut minimum, 1.5 ms untuk sudut tengah, dan 2 ms untuk sudut maksimum.
3. Kembali ke Posisi Awal (Centering): Servo dapat dikendalikan untuk kembali ke posisi tengah atau posisi awalnya dengan mengirimkan pulsa dengan panjang 1.5 ms.

Motor servo merupakan jenis dari motor DC yang dilengkapi gearbox, sensor sudut dan rangkaian kendali. Motor servo mampu mengatur atau menentukan besarnya posisi sudut pada keluaran poros motor, dengan menggunakan sistem kontrol umpan close loop. Komponen yang menyusun motor servo antara lain, potensiometer, sebuah motor dc, rangkaian yang berupa gear dan kontrol. Potensiometer berguna pada motor servo sebagai penentu batas posisi putar pada motor.

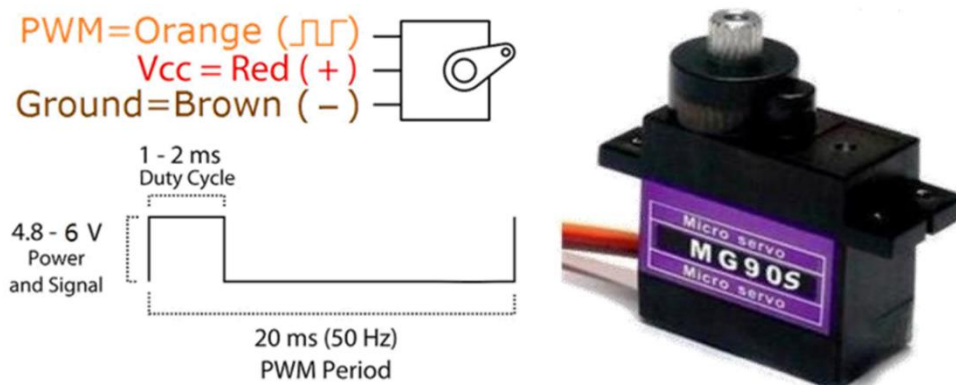


Gambar IV.4. Konstruksi motor servo

Motor servo ada jenis standar yang hanya bergerak di area 180° atau 300° , ada juga jenis kontinyu yang dapat berputar lebih dari 360° namun tidak sensitif terhadap sudut. Servo yang kita gunakan yaitu *Tower Pro Micro Servo MG90S* termasuk motor servo standar yang hanya bisa bergerak 180° . Konfigurasi pin Tower Pro Micro Servo MG90S yaitu :

1. Kabel merah = kutub positif pada baterai / power rail,
2. Kabel coklat = kutub negatif pada baterai / ground,
3. Kabel orange = sinyal masukan pengendali / input control signal (PWM).

Konfigurasi ini ditunjukkan pada gambar IV-2.



Gambar IV.5 Tower Pro Micro Servo MG90S

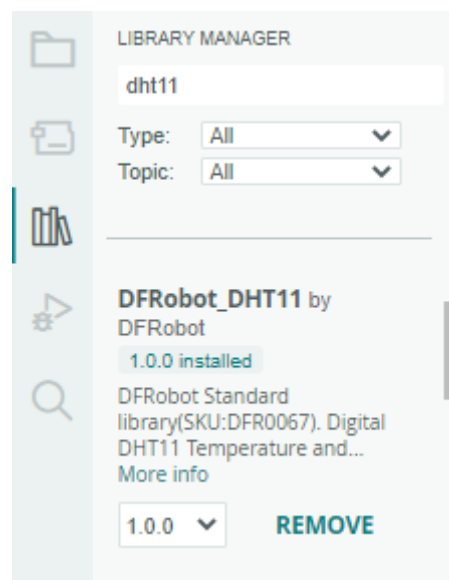
IV.3 Alat dan Komponen yang Digunakan

- 1) Laptop/PC dengan Software IDE Arduino Lastest
- 2) NodeMCU + USB Cable 1 buah
- 3) Kabel Jumper Female-Female 6 buah
- 4) Sensor DHT11 1 buah
- 5) Sensor ultarsonik (HC-SR04) 1 buah
- 6) Servo mikro servo SG90 1 buah

IV.4 Langkah-Langkah Percobaan

IV.4.1 Langkah Persiapan

1. Tersedia IDE Arduino, pastikan sudah terinstal sudah terinstal ESP8266 board, pilih dengan cara:
Tools->Board->ESP8266 board->NodeMCU1.0(ESP-12E Module)
2. Tersedia Laptop/PC, pastikan hardware ESP8266 sudah terinstal. Buka “Device Manager” pada windows anda, dan lihat nomor COM yang terhubung.
3. Instalkan Library DFRobot_DHT11 by DFRobot



IV.4.2 Langkah Workshop

IV.4.2.1 Mengendalikan Intensitas Cahaya Led_Builtin Menggunakan PWM

1. Tanpa ada rangkaian komponen lain, hanya menggunakan Led_Builtin
2. Ketik dan upload program berikut, simpan dengan nama ESP8266_OutputAnalogPWM_Serial:

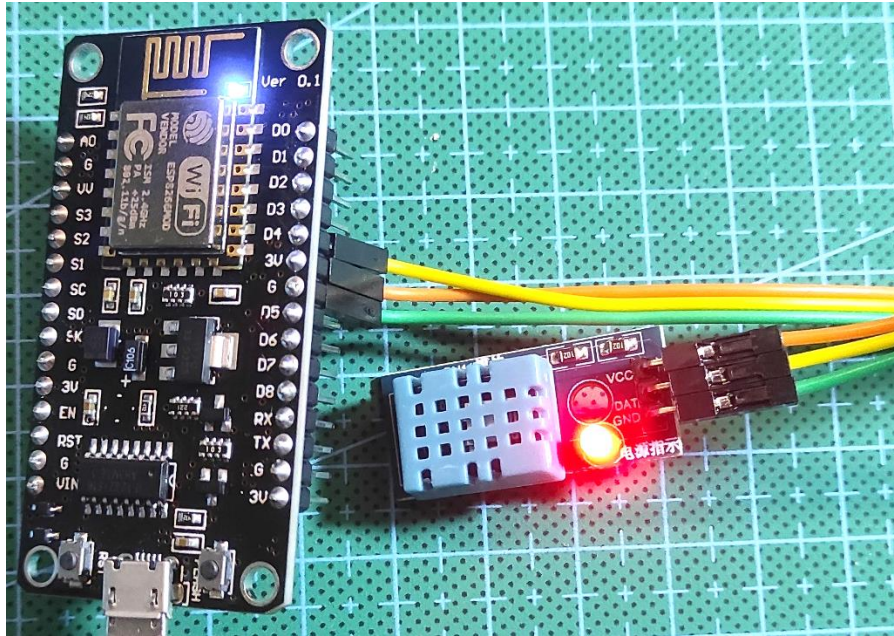
```
int baca, nilai,x; //buat variable baca sensor
String text="";
int led=D4;
void setup() {
  Serial.begin(115200);
  pinMode(led,OUTPUT);
}
void loop() {
  if(Serial.available()){ delay(10); text="";//serial masuk
  while(Serial.available()){
    text+=(char)Serial.read(); // baca isi serial
  }
  nilai=text.toInt(); //ubah data masuk dari text ke angka
  Serial.println(nilai);
  //1023-nilai -> pembalik kondisi LED_Builtin aktif low
  analogWrite(led,1023-nilai);//0=padam,200=redup,1023 Max
  }
  delay(100);
}
```

3. Testing dan Observasi
 - Buka Tools->Serial Monitor atau Ctrl+Shift+M
 - Pada langkah ini seharusnya, data diinputkan 0 sd 1023 pada serial akan mengubah intensitas Led_Builtin, 0=padam, 200=redup, 1023 Max
 - Jika berhasil coba hapus angka 1023- pada “1023-nilai” menjadi “nilai” saja, dan lakukan seperti pada poin sebelumnya, dan amati hasilnya akan terjadi sebaliknya.

IV.4.2.2 Membaca Suhu dan Kelembaban Menggunakan Sensor DHT11

1. Hubungkan Sensor DHT11 ke NodeMCU:

ESP8266	DHT11
D4	DAT (out)
3V	VCC (+)
G	GND (-)



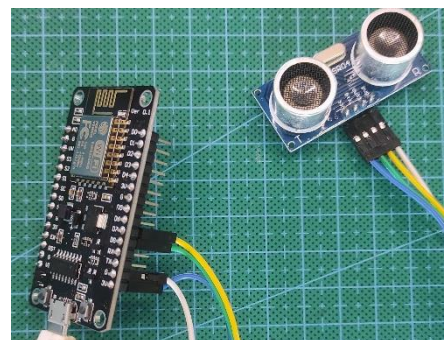
2. Buat file example pada IDE Arduino,
File->Examples->DFRobot_DHT11->ReadDHT11, lalu simpan dengan nama “ESP8266_DHT11”
3. Ubah bagian program berikut dan Upload:

```
#define DHT11_PIN D4 //dari 10 jadi D4
```
4. Testing dan Observasi
 - Buka Tools->Serial Monitor atau Ctrl+Shift+M
 - Maka akan muncul data Suhu dan Kelembaban udara sekitar sensor,
 - Uji suhu pada keluaran fan laptop anda, atau suhu dan kelembaban nafas anda.
5. Pengembangan kasus, silahkan berkreasi.

IV.4.2.3 Contoh Membaca Jarak Menggunakan Sensor Ultrasonic HC-SR04

1. Hubungkan Sensor Ultrasonic HC-SR04 ke NodeMCU:

ESP8266	Ultrasonic
3V	VCC
D8	Trig
D7	Echo
G	Gnd



2. Jika belum ada lib servo, buka Arduino IDE, manage libraries, cari dan install library “NewPing” by Tim Eckel

Pada IDE arduino Buka File->Examples ->NewPing-> NewPingExample

Save as “ESP8266_Ultrasonic”

3. Ubah dan Upload beberapa bagian code program berikut:

```
#include <NewPing.h>
#define TRIGGER_PIN D8 // HC-SR04 pin trigger
#define ECHO_PIN D7 // HC-SR04 pin echo
#define MAX_DISTANCE 400 // Max HC-SR04 at 400-500cm.
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //setup
void setup() {
    Serial.begin(115200);
}
void loop() {
    delay(50); // delay pembacaan
    //Serial.print("Ping: "); //dijadikan komentar tak digunakan
    Serial.print(sonar.ping_cm()); // get distance in cm
    Serial.println(); //bagian ("cm") dihapus
}
```

Mungkin saat upload program akan error bagian connecting:

connecting

jika gagal upload dengan pesan diatas atau pesan sejenisnya, cabut dulu D8 dan D7 saat uploading, pasang kembali jika sudah selesai proses uploadnya.

4. Testing dan Observasi

Buka Tools->Serial Plotter atau Ctrl+Shift+L

Pada langkah ini seharusnya,

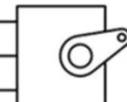
1. Data tampil dalam data skala 0 sd 400 cm,
2. Arahkan sensor pada objek dengan jarak yang diubah-ubah, sehingga terlihat respon pembacaan sensor terhadap perubahan jaraknya.

5. Pengembangan kasus, silahkan berkreasi agar lebih memahami isi programnya.

IV.4.2.4 Contoh Mengendalikan Servo dan setting posisi

1. Hubungkan Servo ke NodeMCU:

ESP8266	Servo
D4	Orange
3V	Merah
G	Coklat

PWM=Orange (□□) 
Vcc = Red (+)
Ground=Brown (-)

2. Pada IDE arduino Buka File->Examples-> Servo->sweep

Edit bagian program berikut:

```
myservo.attach(D4,500,2500); //set pin, sudut 0=500 dan 180=2500
```

3. Testing dan Observasi, buka serial monitor

Perhatikan dan pahami setiap baris programnya, disini ada 2 bagian program, pertama yang memberikan perintah berputar CCW (*Counter Clock Wise* /lawan arah jarum jam) dengan For sudut 0° sd 180°,

```
for (pos = 0; pos <= 180; pos += 1) {
  myservo.write(pos); // memberi perintah sudut 0-180
  delay(15); // delay menunggu sampai menuju posisi tersebut
}
```

dan perintah berputar CW (searah jarum jam) dengan For sudut 180° sd 0°.

```
for (pos = 180; pos >= 0; pos -= 1) {
  myservo.write(pos); // memberi perintah sudut 180-0
  delay(15); // delay menunggu sampai menuju posisi tersebut
}
```

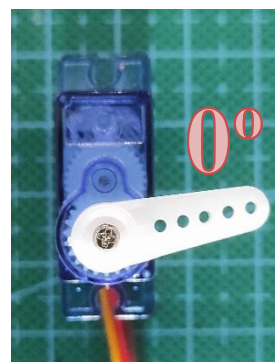
- Selanjutnya setting posisi tuas servo. Ubah agar dalam void loop hanya ada 1 baris program:

```
myservo.write(90); // memberi perintah sudut 90 derajat
```

- Kita pasang tuas servo seperti gambar berikut, usahakan selurus mungkin tanpa memaksakan lurus, pasang baud kecilnya jika tersedia obeng:



Sehingga saat di berikan perintah sudut 180 derajat akan seperti gambar kiri dan saat di berikan perintah sudut 0 derajat akan seperti gambar kanan,



- Pengembangan kasus, silahkan berkreasi agar lebih memahami isi programnya.

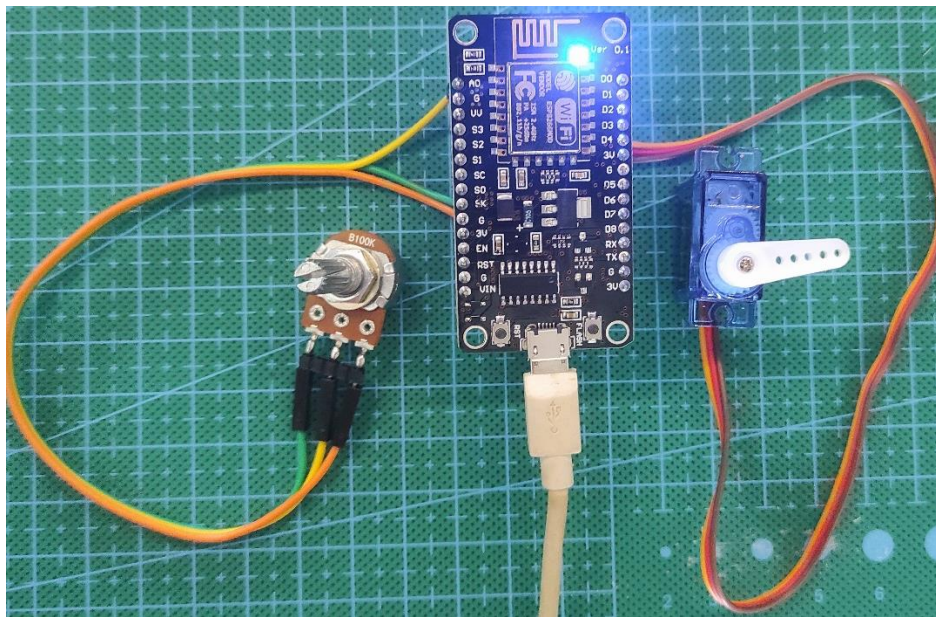
IV.4.2.5 Ujicoba Analog dan Kendali Servo menggunakan Potensiometer

1. Hubungkan Potensiometer dan Servo ke NodeMCU:

ESP8266	Potensiometer
G	1
A0	2
3V	3

ESP8266	Servo
D4	Orange
3V	Merah
G	Coklat

Hati-hati, cek kembali Pin 2 Potensiometer jangan salah pasang !!!!



2. Coba code berikut untuk uji coba sebelum terhubung ke Ubidots

```
#include <Servo.h>
Servo myservo;
int data,deg;
void setup() {
  Serial.begin(115200);
  myservo.attach(D4,500,2500);//set pin dan sudut 0-180
}
void loop() {
  data=analogRead(A0);
  deg=map(data,0,1024,0,180);//ubah skala 0-1023 ke 0-180
  Serial.println(deg);
  myservo.write(deg);
  delay(100);
}
```

Simpan dengan nama ESP8266_Potensio_Servo, sesuaikan arah potensio dengan ubah `map(data,0,1024,0,180)` jadi `map(data,1024,0,0,180)`

3. Testing dan Observasi

- Saat potensiometer diputar, servo ikut berputar mengikuti putaran potensiometer.

IV.4.2.6 Monitoring Analog dan Kendali Servo Melalui Ubidots

1. Masih menggunakan rangkaian sebelumnya (IV.4.2.5)
2. Buka program sebelumnya ESP8266_Ubidots_Potensiometer_Relay dari percobaan III.4.2.3
Save as "ESP8266_Ubidots_Potensiometer_Servo"

Untuk menghubungkan NodeMCU ke Cloud ada 5 hal penting yang perlu di sesuaikan antara Alat kita dengan IoT Platform-nya, diantaranya sbb. :

- a. Nama SSID Wifi,
 - b. Password Wifi
 - c. Token IoT Platform
 - d. Nama Device, umumnya hanya 1 saat latihan
 - e. Nama Variabel sensor/aktuator, ini bisa lebih dari 1.
3. Sesuaikan, ubah/tambahkan beberapa bagian programnya yang di highlight dengan cermat:

```
#include "UbidotsESPMQTT.h"
#include <Servo.h> //tambahkan
#define TOKEN "BBUS-XngSGdvsk8AGhgrm0"//Your Ubidots TOKEN
#define WIFINAME "Sejuk" //Your SSID
#define WIFIPASS "lupalagi" // Your Wifi Pass
Servo servo;
const int led=D4; // hapus
int analog,deg;

//***** Di bagian Auxiliar Functions *****
void callback(char* topic, byte* payload, ....
    text = ""; //tambah ini
    for (int i=0;i<length;i++) {
        text += (char)payload[i]; //tambah ini
        Serial.print((char)payload[i]);
    }
    if ((char)payload[0]=='1'){
        digitalWrite(led, HIGH);
    }
    else{
        digitalWrite(led, LOW);
    }
    Serial.println();
//Bagian atas ini dihapus, diganti menjadi dibawah ini
    Serial.println();
    Serial.println(text);
    deg=text.toInt();
    Serial.println(deg);
```

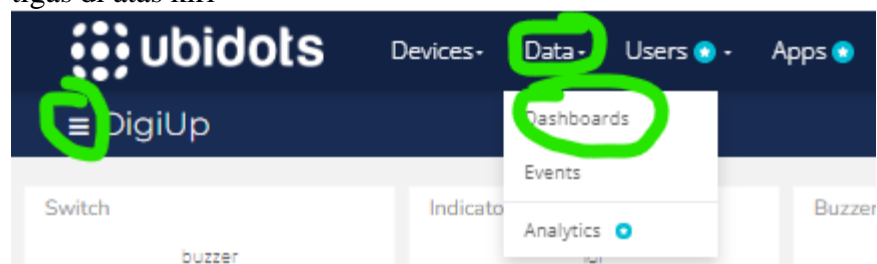
```
servo.write(deg);

//***** Di bagian Main Function *****
void setup() {...
  servo.attach(D4,500,2500); //tambah ini deklarasi servo
  pinMode(led, OUTPUT); // hapus tidak digunakan
  client.ubidotsSubscribe("esp8266_dnb","servo"); //(nama
  device label, nama variable label)

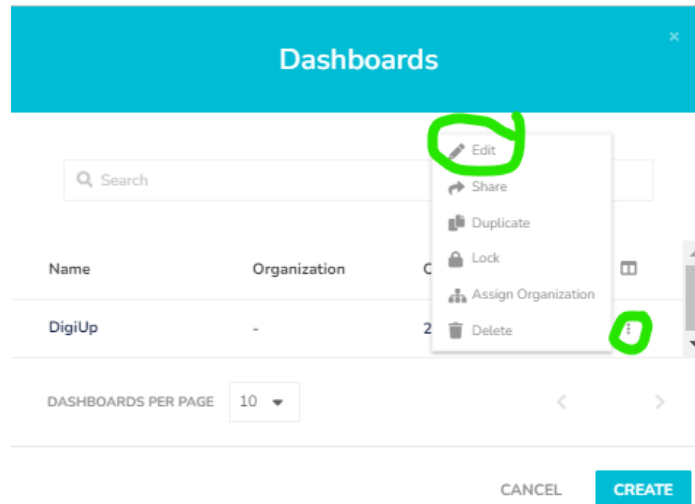
void loop() {
  if(!client.connected()){
    client.reconnect();
    //(nama device label, nama variable label)
    client.ubidotsSubscribe("esp8266_dnb","servo");
  }
  analog = analogRead(A0); //baca analog dari Pin A0
  Serial.println(analog); //kirim data analog ke Serial
  client.add("potensiometer", analog); //data potensiometer
  dari A0
  client.ubidotsPublish("esp8266_dnb"); //max send 4.000
  dots/d (Monitoring)
  client.loop(); //max receive 500.000 dots/d (Control)
  delay(5000); //hemat kuota, 21s=26jam, 10s=12jam, 5s=6jam
}
```

4. Buat aplikasi di bagian Cloud, IoT webserver yang kita gunakan Ubidots.

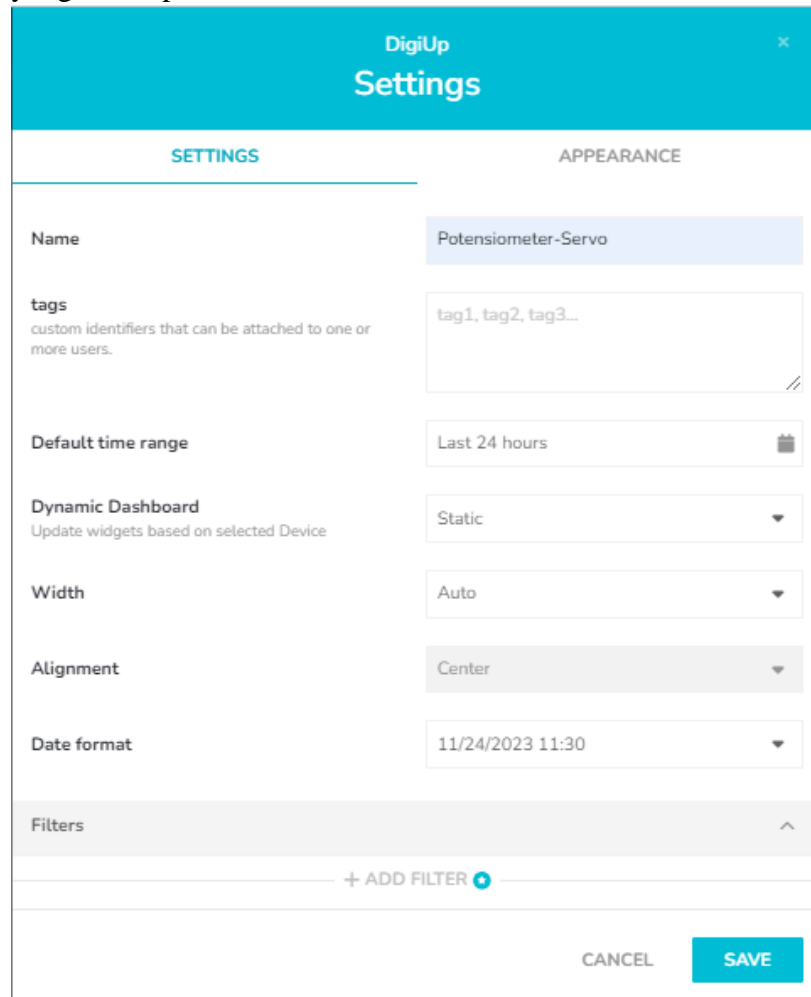
- 1) Device pada akun Ubidots Stem masih menggunakan yang sebelumnya di Part II dan III “esp8266_dnb”.
- 2) Masih menggunakan token sebelumnya
- 3) Pada menu device “esp8266_dnb”:
 - Kita tambahkan variable baru variable baru “servo” untuk potensiometer, dengan cara yang sama pada part II dan III
- 4) Pada menu dashboard modifikasi nama dashboard-nya menjadi “Potensiometer-Servo” dengan cara sbb.:
 - Pada dashboard akan muncul dashboard sebelumnya , click garis tigas di atas kiri



- Klik menu (3 titik) di bagian kanan dari nama dashboard lama, pilih edit



- Lalu ubah nama dashboardnya menjadi “Potensiometer-Servo”, bisa juga mengubah parameter lainnya agar sesuai dengan tampilan yang diharapkan. Lalu tekan SAVE

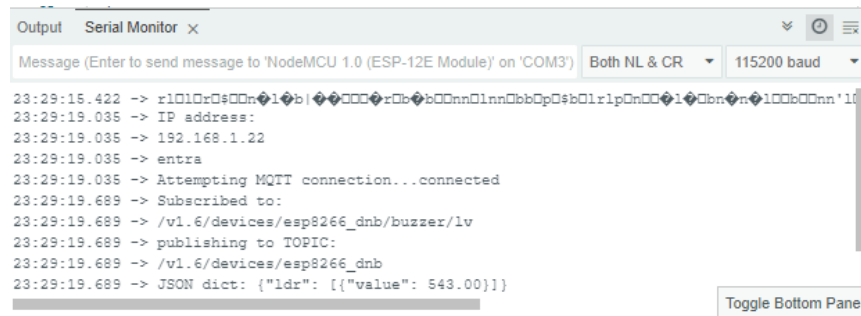


- Selanjutnya kita melihat widgets sebelumnya, hapus semua kecuali “Gauge” dengan add variable “potensiometer”.

- Tambahkan data widget baru “Tank” dengan add variable “potensiometer” dengan cara yang sama seperti part II. Sebelum tekan “save” kita dapat mengubah nama widget dan parameterlainnya pada bagian “Appearance”.
- Tambahkan data widget baru “Slider” dengan add variable “servo” dengan cara yang sama seperti sebelumnya, tidak lupa mengatur appearance data max =180.
- Tambahkan data widget baru “Gauge” dengan add variable “servo” lagi.

5. Testing dan Observasi

- Buka Tools->Serial Monitor atau Ctrl+Shift+M
- Serial plotter, cek apakah berhasil terhubung ke wifi, apakah terhubung ke ubidots. Berikut contoh tampilan jika berhasil terhubung ke wifi dan ubidots



```

23:29:15.422 -> r10l0r0$00n1b|0000r0b000n0l0n0bb0p0b0l1p0n00100b00n'10
23:29:19.035 -> IP address:
23:29:19.035 -> 192.168.1.22
23:29:19.035 -> entra
23:29:19.035 -> Attempting MQTT connection...connected
23:29:19.689 -> Subscribed to:
23:29:19.689 -> /v1.6/devices/esp8266_dnb/buzzer/lv
23:29:19.689 -> publishing to TOPIC:
23:29:19.689 -> /v1.6/devices/esp8266_dnb
23:29:19.689 -> JSON dict: {"ldr": [{"value": 543.00}]}
    
```

- Serial plotter, dan lakukan seperti pada percobaan sebelumnya memutar kira dan kanan pada potensio meter. Akan nampak data baru setiap 5 detik
 - Saat potensiometer dioperasikan perhatikan pada Web Ubidots dan layar Smartphone. Data akan update setiap 5 detik sesuai program kita
 - Servo melalui slider. servo akan merespon setiap 5 detik sesuai program kita.
6. Pengembangan, silahkan program arduinonya save as “..._pengembangan”

LAMPIRAN