

## BAB III

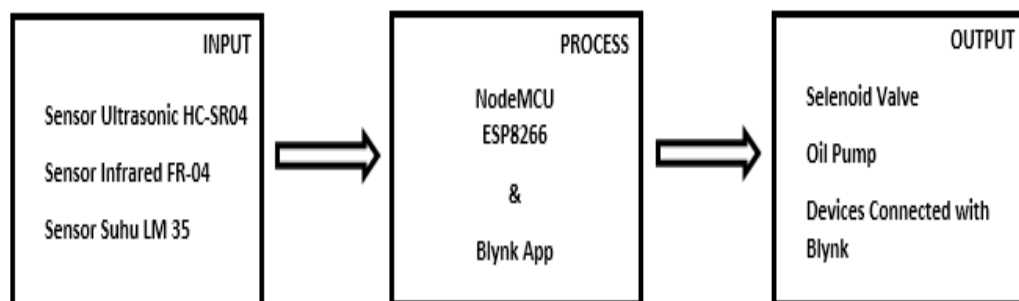
### PERANCANGAN DAN PEMBUATAN ALAT

#### 3.1 Gambaran Umum

*Prototype* Sistem Pelumasan Otomatis pada Mesin *Tunnel Oven* Berbasis *IoT* ini terdiri dari dua perancangan yaitu perancangan perangkat keras dan perancangan perangkat lunak. Secara garis besar perangkat ini berfungsi sebagai sistem *preventive maintenance* pada mesin *tunnel oven* dengan kontrol pompa oli dan *solenoid valve* melalui variabel suhu didalam *oven* untuk mencegah pelumasan pada media pelumas. Data pembacaan sensor akan langsung diproses oleh mikrokontroler ke aplikasi *Blynk* dan tersaji secara *real-time* sehingga sistem ini dapat di *monitoring* pula sistem kerjanya.

#### 3.2 Blok Diagram Sistem

Sebelum menentukan dan membuat perancangan sistem ini, terlebih dahulu dilakukan perencanaan blok diagram yang akan menjadi kerangka acuan dalam pembuatan sistem sesuai dengan yang diinginkan. Adapun blok diagram dari sistem ini terdapat pada gambar 3.1.

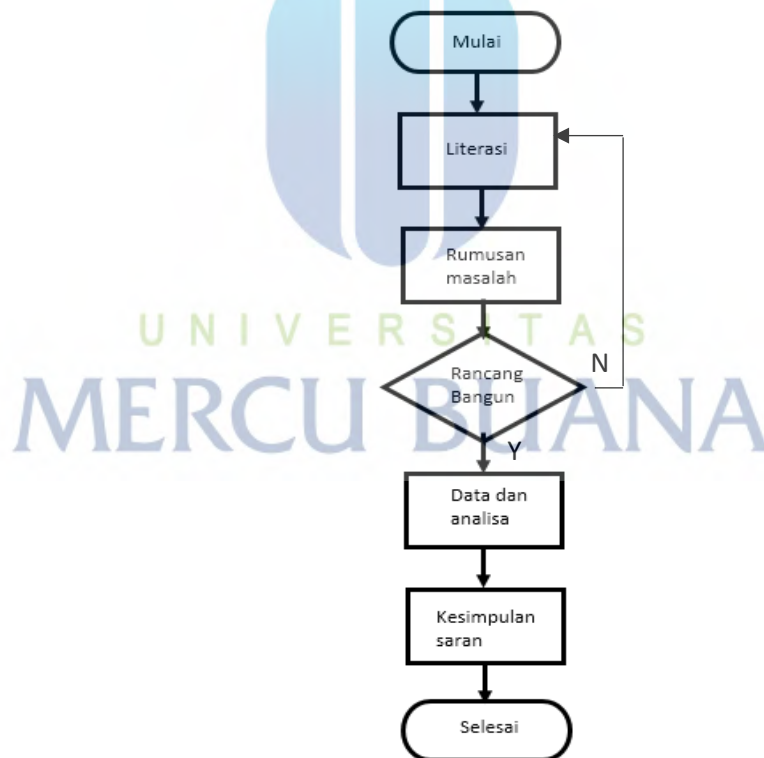


Gambar 3. 1 Diagram alir sistem

Dari gambar 3.1 dapat dijelaskan bahwa sistem dibagi menjadi 3 bagian blok, yaitu *input*, proses, dan *output*. Sensor *ultrasonic*, sensor infra merah, dan sensor suhu merupakan bagian daripada blok masukan (*input*), sementara *solenoid valve*, *oil pump* masuk sebagai blok keluaran (*output*). *Solenoid valve* dan *oil pump* ini terhubung dengan kontak utama *module relay*, serta *smartphone* masuk kedalam blok keluaran juga. Untuk mikrokontroler *nodeMCU ESP 8266* dan perangkat lunak *Blynk* berfungsi sebagai kontroler untuk mengolah data yang diterima dari blok *input*.

### 3.3 Diagram Alir Penelitian

Untuk tahapan dari proses yang akan dilakukan dalam perancangan ini digambarkan dalam diagram alir sebagai berikut :



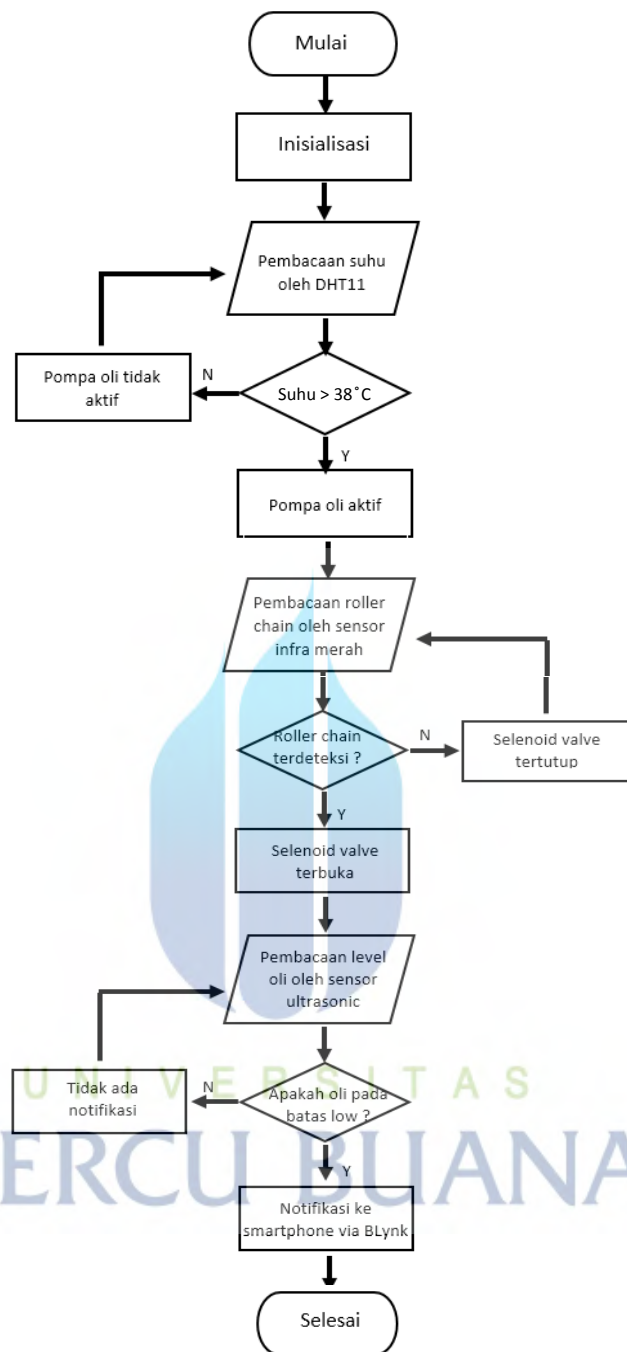
Gambar 3. 2 Diagram alir penelitian

Gambar 3.2 mendeskripsikan keseluruhan proses penelitian dan pada saat rancang bangun tidak didapati hasil yang diinginkan maka proses studi literasi akan dilakukan kembali.

### 3.4 Prinsip Kerja Alat

Sistem Pelumasan Otomatis berbasis *IoT* pada Mesin *Tunnel Oven* ini pada prinsipnya bekerja secara otomatis dengan syarat kondisi pertama yaitu suhu *oven* telah mencapai  $\leq 80^{\circ}\text{C}$  pembacaan ini dideteksi oleh sensor suhu DHT11, namun pada perancangan protipe ini *set point* suhu diubah menjadi syarat kondisi pertama yaitu saat suhu  $> 38^{\circ}\text{C}$ . Kondisi ini akan mengaktifkan *module relay* yang *terminal switch* nya terhubung dengan pompa oli. Setelahnya kondisi kedua sensor inframerah FC-51 akan mendeteksi *roller chain*, setiap pembacaan ini akan memberi sinyal ke pin *digital input module relay* yang *terminal switch*-nya terhubung dengan *solenoid valve* sehingga sejumlah oli akan tersembur ke *roller chain conveyor*. Jika kondisi pertama tidak terpenuhi maka pembacaan sensor infra merah tidak akan *valid* dan proses ini akan terus berjalan sampai *tunnel oven* dinyalakan kembali atau saat suhu  $\leq 38^{\circ}\text{C}$  maka sistem akan berhenti.

Selanjutnya pada bagian tangki oli sensor *ultrasonic* HC-SR04 akan membaca *level* ketinggian oli secara *continue* dan akan memberikan notifikasi pada *devices* hanya jika oli telah mencapai batas *level low* sesuai yang telah ditentukan. *NodeMCU* ESP 8266 yang telah terhubung dengan jaringan internet akan men-transmisikan *data* kepada *devices* yang telah terinstal *Blynk App*. Notifikasi yang ditampilkan berupa *push notification* dan data pembacaan dari beberapa sensor yang digunakan terlihat dalam bentuk grafik ataupun angka pada *smartphone*. Gambar 3.3 menunjukkan diagram alir prinsip kerja sistem otomatis pelumasan pada mesin *Tunnel oven*.



Gambar 3.3 diagram alir prinsip kerja sistem

Gambar 3.3 merupakan gambar dari diagram alir keseluruhan proses rancang bangun sistem otomatis pelumas rantai pada mesin *tunnel oven* berbasis *IoT* dari mulai pembacaan sensor sampai dengan masuknya notifikasi ke *smartphone*.

### 3.5 Perancangan Sistem Pelumasan Otomatis

Untuk merancang dan membuat prototipe sistem pelumasan otomatis berbasis *IoT* pada mesin *tunnel oven* ini dibutuhkan beberapa peralatan dan komponen-komponen diantaranya sebagai berikut :

Peralatan :

1. Obeng +/-
2. Tang potong/kombinasi
3. Lem
4. Laptop untuk pemrograman Arduino IDE
5. Kabel USB untuk proses *upload*

Untuk komponen-komponen nya sebagai berikut :

1. NodeMCU ESP8266
2. Sensor *module Ultrasonic* HC-SR 04
3. Sensor *module Infrared* FC – 04
4. Sensor suhu DHT11
5. Modul *relay 2 channel*
6. *Oil pump* 12 VDC
7. *Solenoid valve* 24 VDC
8. *Power supply* 12 VDC
9. *Power supply* 24 VDC
10. *Oil tank*
11. *Oil tube* 6mm
12. Pipa tembaga ¼"
13. Media pelumas
14. Kabel *jumper*

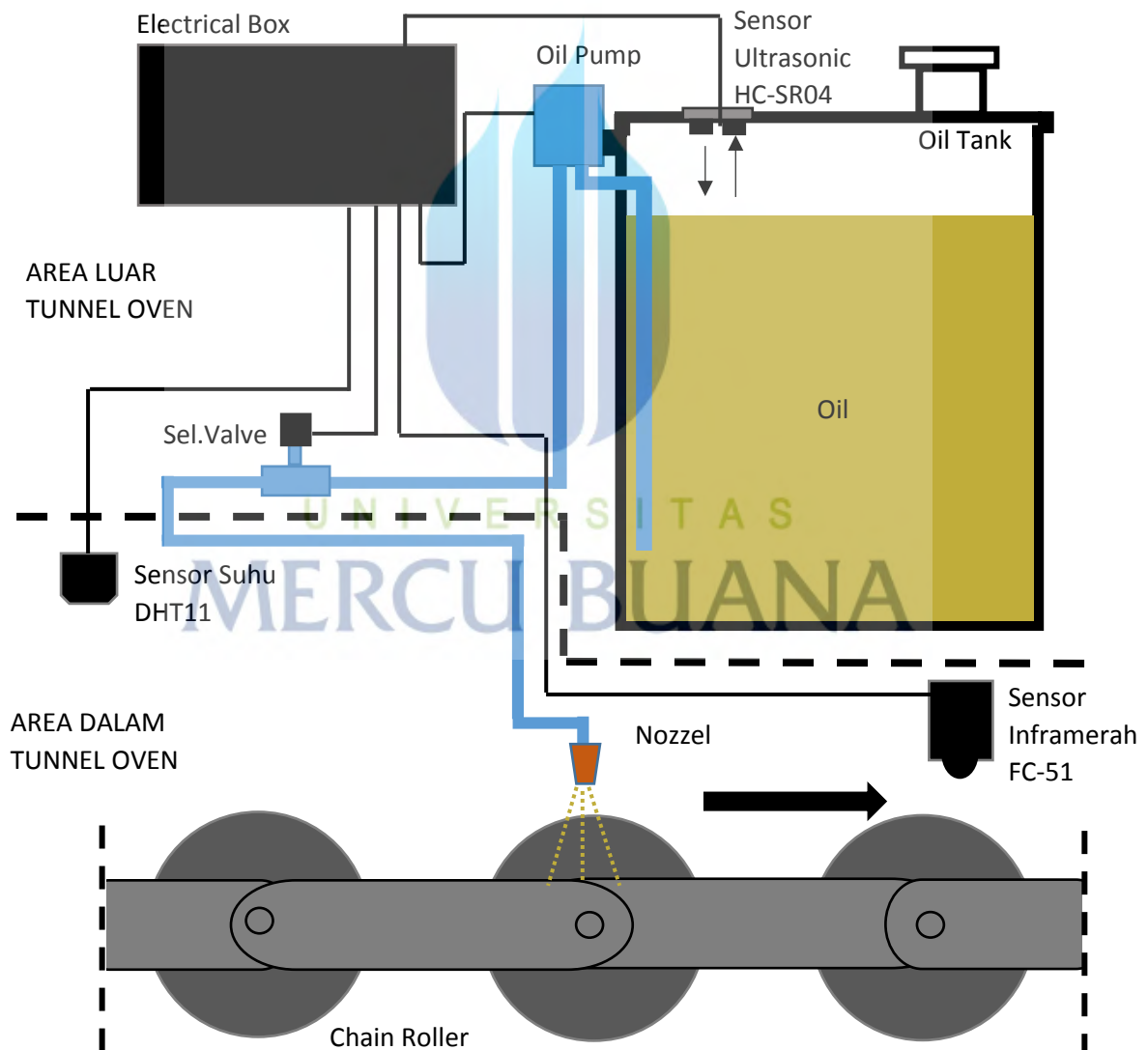
Perancangan ini terbagi menjadi dua bagian yaitu perancangan perangkat keras (mekanikal dan eletrikal) dan perancangan perangkat lunak.

### 3.6 Perancangan Perangkat Keras

Pada bagian ini akan menjelaskan mengenai perancangan mekanikal dan elektrikal dari perancangan sistem yang akan dibuat.

#### 3.6.1 Perancangan Mekanikal

Perancangan mekanikal adalah pembuatan kerangka alat dari komponen-komponen yang telah dibahas sebelumnya. Perancangan ini berbentuk *prototype* yang dibuat sedemikian rupa sesuai dengan keadaan aslinya di lapangan.



Gambar 3. 4 Sketsa sistem otomatis pelumas rantai

Pada gambar 3.4 menjelaskan posisi dari masing-masing sensor serta komponen-komponen yang digunakan. Sensor suhu DHT11 diletakkan di bagian dalam mesin *Tunnel oven* untuk mendeteksi suhu, normalnya saat suhu  $< 80^{\circ}\text{C}$  sensor akan men-*trigger module relay* untuk pompa oli, namun dalam prototipe perancangan ini dibuat saat suhu  $> 38^{\circ}\text{C}$  maka pompa oli akan aktif. Kemudian *module Sensor infrared FC-51* dirancang sedemikian rupa agar membaca *roller chain conveyor* dan menjadikan *roller-roller* tersebut satu-satunya objek yang bisa dideteksi oleh sensor lalu mengaktifkan *module relay* untuk *solenoid valve* dan dengan peletakan *output nozzle solenoid* berada diatas *roller chain* maka pada saat yang bersamaan akan menyemprotkan sejumlah oli pada *roller chain*. Sistem akan terus bekerja sampai sensor suhu membaca suhu  $\leq 38^{\circ}\text{C}$ .

Untuk sensor *ultrasonic HC-SR04* diletakan pada bagian atas *oil tank*. Prinsip pengukuran *level* ini adalah dengan mengukur jarak antara *transmitter* pada *module HC-SR04* dengan ujung permukaan minyak pelumas di dalam tanki. Semakin besar jarak keduanya, maka menunjukkan *level* media pelumas semakin rendah. Demikian pula sebaliknya, semakin pendek jarak keduanya, maka menunjukkan *level* semakin tinggi. Pada titik inilah terjadi kegiatan pertukaran data melalui internet, apabila *level* oli rendah  $< 1$  liter maka sistem ini akan memberikan notifikasi ke *smartphone operator*.

### 3.6.2 Perancangan Elektrikal

Dalam perancangan kelistrikan *prototype* sistem pelumas otomatis pada mesin *tunnel oven* berbasis *IoT* ini harus dilakukan beberapa tahapan, agar alat berfungsi sebagaimana mestinya. Setelah melakukan perancangan mekanik maka untuk merealisasikan sistem yang telah dibuat pada blok diagram perlu dilakukan perancangan kelistrikan. Adapun yang menjadi perancangan kelistrikan meliputi beberapa bagian, yaitu :

#### a. Rangkaian *Input*

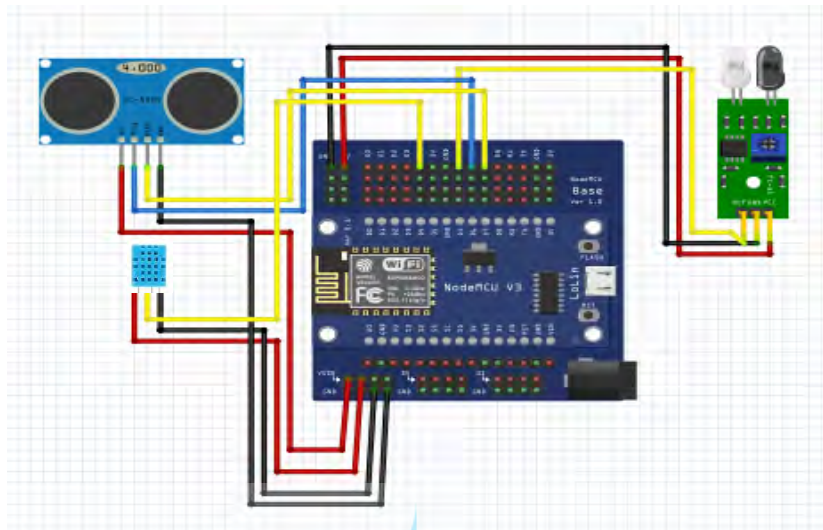
Rangkaian *input* adalah sebuah rangkaian yang meliputi *sensor-sensor* yang digunakan dalam perancangan ini yaitu *sensor infrared FC-51*, *sensor* suhu DHT11 dan sensor *ultrasonic HC-SR04*. Rangkaian ini berfungsi memasukan nilai atau data yang terdeteksi oleh *sensor-sensor* tersebut

yang kemudian akan diproses oleh mikrokontroler NodeMCU ESP8266 sebelum menjadi sebuah sistem. Adapun yang menjadi bagian dari rangkaian *input* yaitu :

- Sensor FC-51, berfungsi untuk membaca dan mengetahui objek yang ada di depan sensor dalam hal ini *roller chain conveyor*. Sensor ini memiliki 3 kaki, untuk kaki *VCC* ditandai dengan kabel berwarna merah dan terhubung pada pin 3.3 V. Kaki *GND* ditandai dengan kabel berwarna hitam dan terhubung pada pin *GND*. Untuk kaki *OUT* atau data ditandai dengan kabel berwarna kuning dan terhubung pada pin D5 (GPIO14) pada mikrokontroler nodeMCU.
- Sensor DHT11, berfungsi untuk membaca suhu pada bagian dalam mesin *tunnel oven*. Sensor ini memiliki 3 kaki, untuk kaki *VCC* ditandai dengan kabel berwarna merah dan terhubung dengan *power* 5 V. Untuk kaki *GND* ditandai dengan kabel berwarna hitam dan terhubung pada pin *GND*. Untuk kaki *OUT* atau data ditandai dengan kabel berwarna kuning dan terhubung dengan pin D2(GPIO4) pada mikrokontroler nodeMCU.
- Sensor HC-SR04, sensor yang dalam perancangan ini berfungsi untuk mendeteksi *level* oli pada tangki. Sensor ini memiliki 4 kaki, untuk kaki *VCC* ditandai dengan kabel berwarna merah dan terhubung dengan pin 5 V. Untuk kaki *GND* ditandai dengan kabel berwarna hitam dan terhubung dengan pin *GND*. Untuk kaki Trig (*trigger*) ditandai dengan kabel berwarna biru dan terhubung dengan pin D6 (GPIO12). Sedangkan untuk kaki *ECHO* ditandai dengan kabel berwarna kuning dan terhubung dengan pin D7 (GPIO13).

Adapun *wiring diagram* sensor-sensor ini dengan mikrokontroler nodeMCU ESP 8266 dapat dilihat pada gambar 3.5.





Gambar 3. 5 *Wiring diagram rangkaian input*

(Sumber Fritzing.com, 2020)

Tabel 3. 1 Pin Sensor pada NodeMCU

Jenis sensor	Pin sensor				
	VCC	GND	ECHO	TRIG	OUT
FC - 51	3.3v	GND			GPIO14(D5)
DHT11	5v	GND			GPIO2(D4)
HC-SR04	5v	GND	GPIO6(D12)	GPIO7(D13)	

Tabel diatas merupakan tabel yang berisi pin yang digunakan dalam perancangan rangkaian *input* untuk keperluan catu daya, pin *trigger* dan *echo*, dan pin *out*.

b. Rangkaian *Output* (Aktuator)

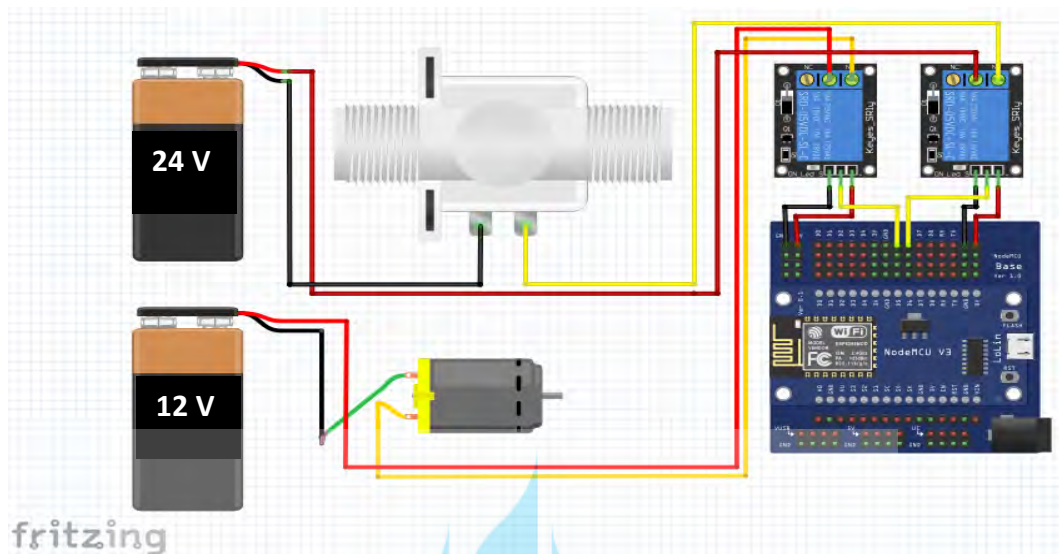
Rangkaian *output* dibuat agar data yang telah diproses oleh mikrokontroler dapat diterima oleh blok *output* dengan baik. Rancangan ini meliputi beberapa komponen yang digunakan sebagai *output* seperti *module relay*, pompa oli, dan *solenoid valve*. Adapaun komponen-komponen yang menjadi bagian dari rangkaian *output* yaitu :

- *Module relay 2 channel*, Modul ini berfungsi untuk menghubungkan komponen lain yang bekerja pada tegangan yang lebih besar. *Module* ini memiliki 3 kaki, untuk kaki *VCC* ditandai dengan kabel berwarna

merah terhubung dengan pin 5 V. Untuk kaki *GND* ditandai dengan kabel berwarna hitam dan terhubung dengan pin *GND*. Untuk kaki *OUT* ditandai dengan kabel berwarna kuning, untuk *channel* pertama terhubung dengan pin D1 (GPIO5) dan *channel* kedua terhubung dengan pin D2 (GPIO4).

- *Solenoid valve*, Gaya elektromagnetik ini dihasilkan oleh koil yang bekerja pada tegangan 24 V. *Solenoid valve* ini memiliki 2 kaki, kaki positif ditandai dengan kabel berwarna kuning dan terhubung dengan kontak *NO* (*normally open*) dari *relay channel* pertama. Kaki negatif ditandai dengan kabel berwarna hitam dan terhubung langsung dengan 0 V pada *power supply* 24 V.
- Pompa oli, adalah mesin *fluida* yang berfungsi untuk memindahkan *fluida* melalui pipa dari satu tempat ke tempat yang lain dalam perancangan ini *fluida* yang digunakan adalah oli/pelumas dan pompa ini bekerja pada tegangan 12 V. *Oil pump* memiliki 2 kaki, kaki positif ditandai dengan kabel berwarna kuning dan terhubung pada kontak *NO* (*normally open*) dari *relay channel* kedua. Kaki negatif terhubung langsung dengan 0 V pada *power supply* 12 V.

Adapun *wiring diagram* dari komponen-komponen ini dengan mikrokontroler nodeMCU ESP 8266 dapat dilihat pada gambar 3.6



Gambar 3. 6 *Wiring diagram rangkaian output*

(Sumber : Fritzing.com, 2020)

Tabel 3. 2 Tabel pin aktuator

Jenis Komponen	Pin modul		
	VCC	GND	INPUT
<i>Relay module 1 (pompa oli)</i>	5v	GND	GPIO5(D1)
<i>Relay module 2 (Sel.valve)</i>	5v	GND	GPIO4(D2)

Tabel diatas menjelaskan pin yang digunakan dalam perancangan rangkaian *output* untuk keperluan catu daya, dan masuka *relay*.

### 3.7 Perancangan Perangkat Lunak

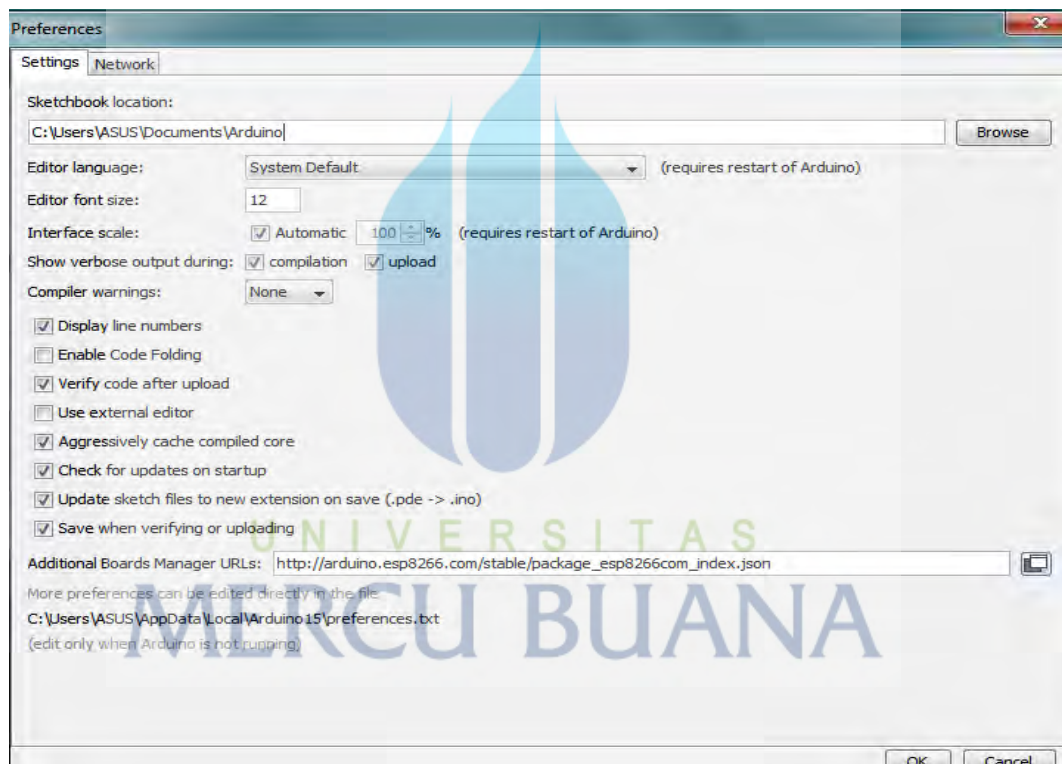
Perancangan perangkat lunak atau pemrograman adalah aktifitas memasukan suatu informasi atau perintah (*coding*) kedalam sebuah mikrokontroler, dengan informasi ini sebuah sistem dapat bekerja sesuai dengan yang diharapkan. Pemrograman pada perancangan sistem ini menggunakan perangkat lunak arduino IDE.

Pada dasarnya perangkat lunak Arduino IDE dikhususkan untuk *board arduino series* seperti arduino uno, arduino mega, dan masih banyak lagi. Namun saat ini pada menu pilihan (*preferences*) dimungkinkan menambahkan *board manager* jenis lain seperti NodeMCU, WeMos dan lain sebagainya dengan

menggunakan bahasa pemrograman berbasis bahasa C pada perangkat lunak arduino IDE.

### 3.7.2 Menambahkan NodeMCU Board Manager

Untuk menambahkan *board manager* pilih *tab File* lalu pilih menu *preferences* kemudian pada bagian *Additional boards manager URLs* ditambahkan *URL* yang menyatakan sebuah fungsi *board manager* untuk NodeMCU ESP 8266 yaitu [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json).



Gambar 3. 7 Preferences untuk NodeMCU ESP8266

Setelah itu pada *tab sketch* pilih *include library* lalu pilih *manage libraries*, pada kotak *filter* ketika ESP 8266 by ESP community lalu tunggu proses *download*-nya selesai. Setelah itu pilih *tab tools* dan pada pilihan *board* pilihan *generic esp8266 module* sudah tersedia. Setelah sukses menambahkan *NodeMCU board manager* ke Arduino IDE selanjutnya melakukan persiapan lain pada menu *Tools* juga seperti pada pilihan port telah disesuaikan dengan *port USB* yang kita gunakan untuk proses *uploading*. Selanjutnya yang harus dilakukan yaitu inialisasi *library*

lain, *library* berisikan fungsi *main program basic* yang hendak kita gunakan dalam sebuah perancangan, seperti penggunaan *sensor*, *servo*, *lcd*, dan lain sebagainya.

### 3.7.2 Pemrograman NodeMCU ESP 8266 Dengan Blynk App

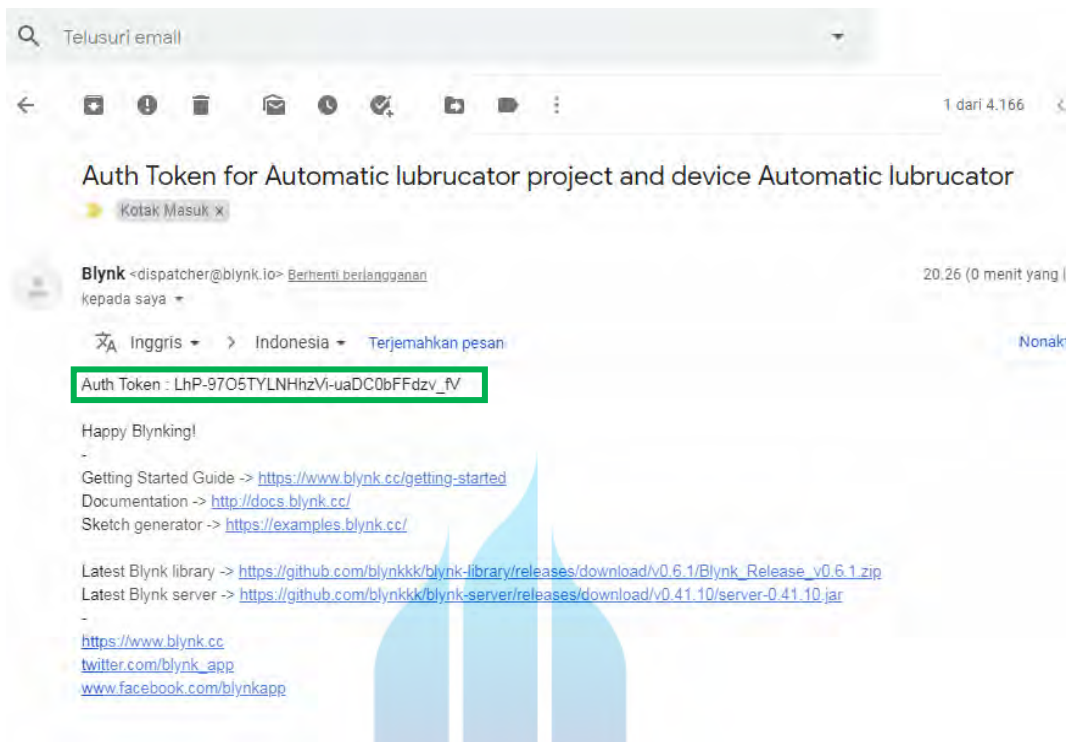
Perrograman untuk koneksi App Blynk dan rangkaian dapat dilihat pada gambar 3.8. Alat ini dirancang untuk dapat menampilkan visual grafik hasil pembacaan data dari sensor-sensor yang digunakan dalam perancangan seperti suhu dan pembacaan *level* oli pada tanki. Data yang telah diproses mikrokontroler nodeMCU dikirimkan oleh *module* ESP8266 yang sudah *built-in* pada *board* via *jaringan internet/Wifi* ke *server* milik Aplikasi *Blynk* dan diterima oleh *Smartphone Android*.



Gambar 3. 8 Koneksi *NodeMCU* dan *Smartphone Android*

Agar mikrokontroler *nodeMCU* dapat terhubung dengan Aplikasi Blynk yang ter-*install* di *smartphone Android* harus memiliki nomor autentikasi token, *SSID* dan *Password wifi* melalui *program* yang di-*upload* pada mikrokontroler *nodeMCU*. Nomor autentikasi token ini didapat ketika pertama kali membuat *project* pada Aplikasi *Blynk* dan akan dikirim *via email* yang didaftarkan, contoh seperti gambar dibawah ini :





Gambar 3. 9 Kode autentikasi

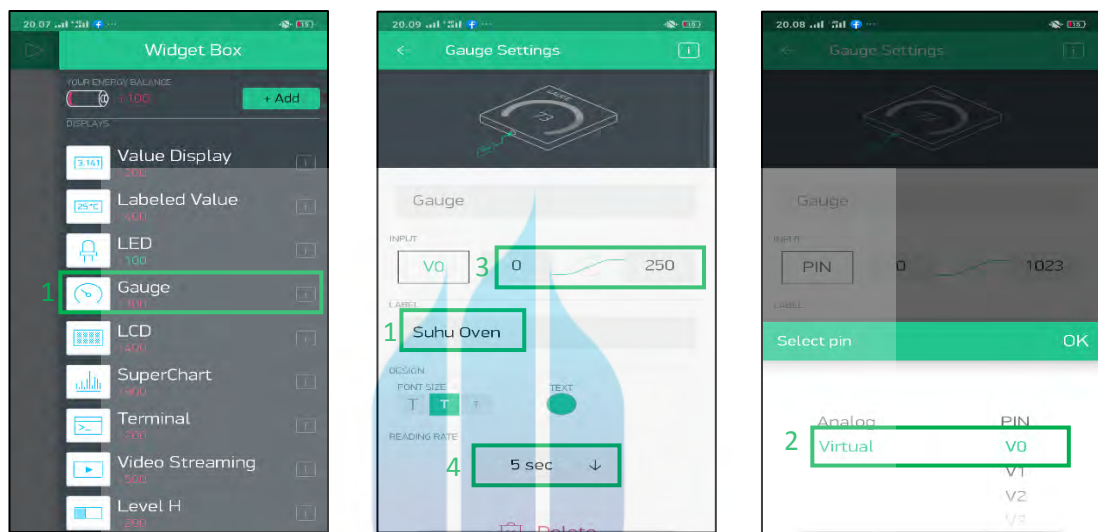
Selain nomor Token dibutuhkan *SSID* dan *password* dari koneksi *wifi* yang akan digunakan dalam program ini. Berikut program untuk meng-*input* nomor token, *SSID* dan *Password Wifi* :

```
char auth[] = "LhP-97O5TYLNHhzVi-uaDC0bFFdzv_fV";
//Kode Autentikasi
char ssid[] = "realme 3"; //Wifi name
char pass[] = "1122334455"; //Wifi Password
```

### 3.7.3 Pemrograman *Blynk App* Untuk Sistem Pelumas Otomatis

Pemrograman ini dilakukan untuk mengatur variabel apa saja yang ingin ditampilkan pada *devices*. Untuk variabel pada perancangan ini yang ditampilkan adalah *gauge* untuk penampil suhu, grafik pembacaan suhu oleh sensor suhu DHT11 dan *LCD display* untuk *level* Tangki oli. Untuk variabel suhu dilakukan dengan tata cara seperti berikut ini :

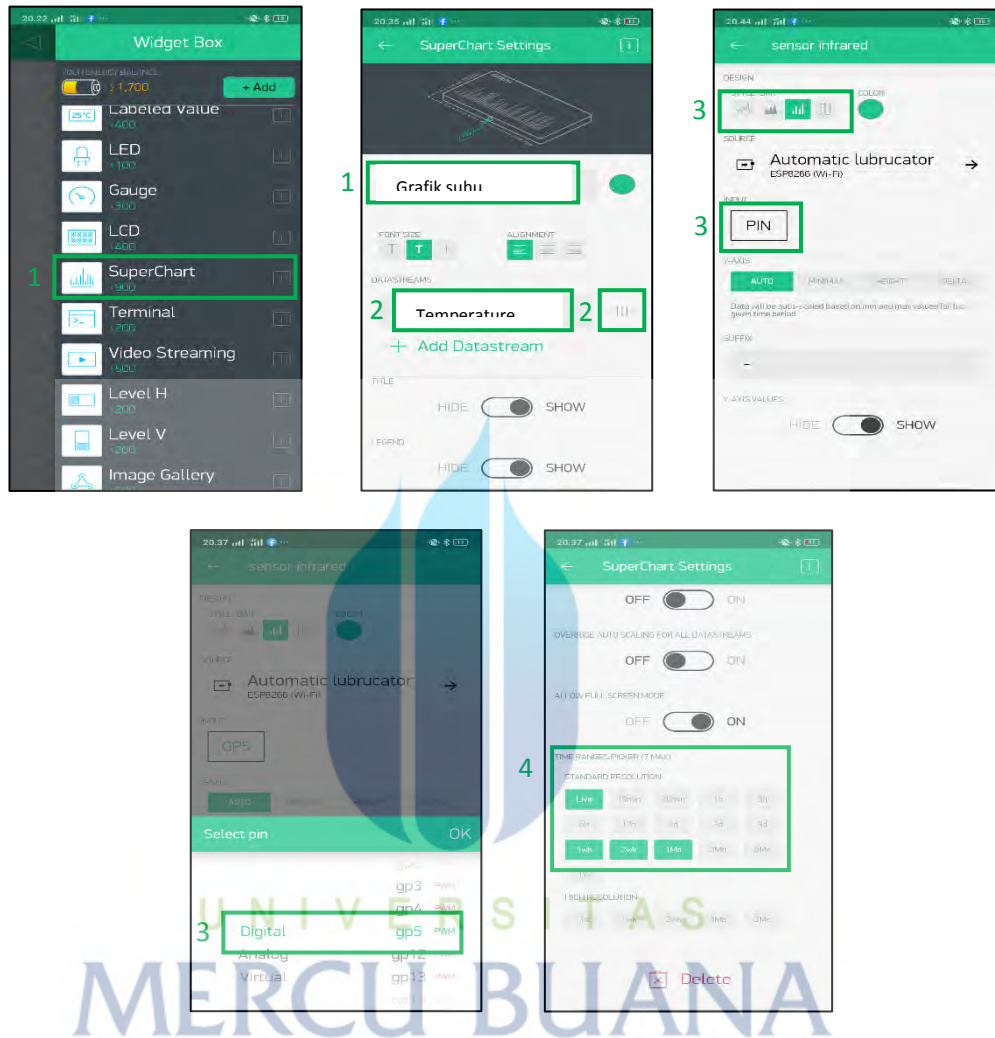
1. Pada *widget box* pilih *Gauge* dan ketik *variabel* sesuai dengan *script* yang dibuat yaitu “Suhu Oven”.
2. Tentukan Pin *Virtual* sesuai *script*, Pin V0 untuk variabel “Suhu Oven”
3. Tentukan *range* pembacaan suhu.
4. Tentukan *delay 5 sec* yang bertujuan untuk *me-refresh* grafik per 5 detik.



Gambar 3. 10 Konfigurasi untuk display suhu

Gambar diatas menjelaskan tata cara menampilkan *gauge* pada aplikasi blynk. Semua variabel ini dikonfigurasi secara sistem melalui fitur pin *virtual* pada aplikasi *Blynk* Selain itu untuk variabel pembacaan suhu juga ditampilkan dalam bentuk grafik yaitu dengan tata cara seperti berikut ini :

1. Pada *widget box* pilih *Superchart* dan ketik *variabel* sesuai dengan *script* yang dibuat yaitu “Grafik suhu” .
2. Pada menu *datastreams* ketikan nama data yang ingin dimasukkan, contoh *temperature*, lalu klik *icon setting* di samping nama ini.
3. Pada konfigurasi *superchart* tentuka bentuk grafik yang ingin ditampilkan, lalu tentukan *virtual pinsesuai* dengan *script* yaitu V6.
4. Tentukan *time range picker* yaitu resolusi pengambilan data terhadap waktu pada contoh dipilih *live*, *1 week*, *2 week* dan *1 month*.



Gambar 3. 11 Konfigurasi untuk grafik pembacaan sensor suhu

Gambar diatas menjelaskan tata cara menampilkan grafik suhu pada aplikasi blynk menggunakan opsi *superchart*. Semua variabel ini dikonfigurasi secara sistem melalui fitur pin *virtual* pada aplikasi *Blynk*. Adapun untuk variabel *level* pada tagki oli dilakukan dengan cara seperti berikut ini :

1. Untuk *LCD* Pada *widget box* pilih *LCD*.
2. Ada 2 jenis *input* pada *LCD setting*, pada perancangan ini *input* yang di pilih adalah *advamce*, lalu untuk isi nya teksnya dikonfigurasi melalui *script* pemrograman.



3. Tentukan *virtual* pin yang disesuaikan pada script pemrograman misal V15.
4. Kemudian untuk notifikasi pada *widget box* pilih *notification* lalu buat perintah *script*-nya.
5. Lalu tentukan apakah akan memberi notifikasi saat sistem *offline* jika iya tentukan waktu responnya, lalu tentukan *level* prioritas notifikasi sistem nya.



Gambar 3. 12 Konfigurasi *LCD* dan notifikasi untuk tangki oli

Gambar diatas menjelaskan tata cara menampilkan *LCD* untuk menampilkan *level* oli pada aplikasi blynk. Semua variabel ini dikonfigurasi secara sistem melalui fitur pin *virtual* pada aplikasi *Blynk*.

### 3.7.4 Pemrograman Sistem Pelumas Otomatis

Sistem ini dimulai dengan pembacaan suhu yang di *input* oleh sensor DHT11 dan mengaktifkan *relay channel* pertama (*oil pump*). Kaki *OUT* sensor ini terhubung dengan pin GPIO4(D2) pada mikrokontroler *nodeMCU*. Berikut ini merupakan contoh *script* pemrogramannya :

```
void sendSensor()//fungsi pengiriman data dari sensor ke
Blynk
{
    float t = dht.readTemperature();// tipe data
    pembacaan suhu
    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT
        sensor!");//mengecek pembacaan apakah terjadi
        kegagalan atau tidak
        return;
    }
    Blynk.virtualWrite(V6, t);//Virtual pin pembacaan
    suhu pada blynk
}

void loop()
{
    Blynk.run();//menjalankan fungsi blynk
    timer.run();//menjalankan fungsi timer
    float t = dht.readTemperature();//pembacaan data
    suhu
    if (t > 38) { //fungsi jika suhu diatas 38 derajat
        digitalWrite(5, HIGH);//relay 1 aktif
    } else {
        digitalWrite(5, LOW);//relay 1 tidak aktif
    }
}
```

Kemudian sistem ini dilanjutkan dengan pembacaan objek oleh sensor infra merah berupa *roller chain conveyor*. Kaki *OUT* sensor infra merah terhubung dengan pin D5 (GPIO14) merupakan pin dalam pada mikrokontroler *nodeMCU* yang secara umum dapat digunakan sebagai pin *input* maupun *output* berdasarkan *script* pemrogramannya. Dengan logika “*High*” dan “*Low*”, dalam sistem ini berfungsi untuk mengendalikan komponen *solenoid valve*. Berikut contoh *script* pemrogramannya :

```

void loop()
{
  Blynk.run();//menjalankan fungsi blynk
  timer.run();//menjalankan fungsi timer
  bool sensorstate = digitalRead(IRSensor);

  if((t> 38) && (sensorstate == LOW)){//fungsi jika
  suhu diatas 38 derajat dan ir sensor bernilai 0
  delay(3000);
  digitalWrite(4, HIGH);//relay 2 aktif
  } else {
  digitalWrite(4, LOW);//relay 2 tidak aktif
  }
}

```

Selanjutnya perancangan ini memiliki sistem *monitoring* pada tangki oli-nya dimana sistem akan memberikan *push notification* ke *smartphone* yang terinstal aplikasi *Blynk* di dalam nya. Sistem *monitoring* ini menggunakan sensor *ultrasonic* untuk membaca *level* oli yang tersisa pada tangki. Berikut contoh *script* pemrogramannya :

```

void sendSensor()//fungsi pengiriman data dari sensor ke
Blynk
{
  //SET UP NOTIFICATION FOR LOW LEVEL OIL
  if (vol < 1){
    Blynk.notify("Oil in Low level, please Refill
the Oil :*");
  }
}

```

```

void setup()
{
  Serial.begin(9600);//kecepatan pengiriman dan
penerimaan data melalui port serial
  Blynk.begin(auth, ssid, pass);// konfirmasi kode
auth, ssid dan password jaringan

  pinMode(TRIGGERPIN, OUTPUT);
  pinMode(ECHOPIN, INPUT);

  lcd.clear(); //Use it to clear the LCD Widget
  lcd.print(1, 0, "Value in Litre"); // use:
(position X: 0-15, position Y: 0-1, "Message you

```

```

want to print") // Please use timed events when
LCD printing in void loop to avoid sending too
many commands // It will cause a FLOOD Error, and
connection will be dropped

    timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run();//menjalankan fungsi blynk
  timer.run();//menjalankan fungsi timer

  lcd.clear();
  lcd.print(1, 0, "Value in Litre"); // use:
(position X: 0-15, position Y: 0-1, "Message you
want to print") long duration, distance;

  digitalWrite(TRIGGERPIN, LOW);
  delayMicroseconds(3);
  digitalWrite(TRIGGERPIN, HIGH);
  delayMicroseconds(12);
  digitalWrite(TRIGGERPIN, LOW);
  duration = pulseIn(ECHOPIN, HIGH);
  distance = (duration/2) / 29.1;
  vol = ((28 - distance)*200)/1000;

  Serial.print("distance");
  Serial.println(distance);
  Serial.print(vol);
  Serial.println("L");

  lcd.print(7, 1, vol);
  delay(1000);//delay setiap 1 detik

```

Setelah penulisan *coding* program arduino selesai, maka berikutnya adalah meng-*upload/compile* program tersebut ke nodemcu dengan langkah-langkah sebagai berikut :

1. Simpan *file* tersebut dengan masuk ke menu “*File*” > “*Save*”. Beri nama *filenya*.

2. Klik tombol “*Verify*” untuk memverifikasi program tersebut guna mengetahui apakah penulisan sudah benar ataukah ada kesalahan. Jika terdapat kesalahan dalam pemrograman pada bagian bawah jendela *sketch* arduino akan muncul keterangan berupa *error list* dan jika pemrograman sudah benar maka akan ada tulisan “*Done verifying*” apabila proses *verify* berhasil.
3. Hubungkan arduino *board* ke laptop dengan kabel data USB. Klik tombol “*Complie*” dan tunggu sampai proses *upload* program selesai.
4. Setelah proses *compile* selesai, maka led pada nodeMCU akan *blinking* dan nodeMCU sudah bersungsi sesuai dengan program yang di*upload*.

