

## **BAB 2**

### **DASAR TEORI DAN TINJAUAN PUSTAKA**

#### **2.1 Teori Terkait**

##### **2.1.1 Kelinci**

Kelinci adalah salah satu hewan yang menggemaskan dan cocok untuk dijadikan hewan peliharaan. Selain untuk konsumsi, kelinci dapat ditenakkan menjadi hewan peliharaan atau hewan hias. Bulunya yang lembut dan bentuknya yang menggemaskan tentu saja akan menarik perhatian masyarakat yang suka memelihara hewan ternak.

Kelinci merupakan hewan yang memiliki daya adaptasi tubuh yang relatif tinggi. Oleh karena itu hampir setiap negara di dunia memiliki peternakan kelinci (Alex S, 2014).

##### **2.1.2 Pemberian Pakan**

Ada dua cara pemberian pakan pada kelinci hias peliharaan, yaitu *hand feeding* dan *self feeding*.

###### **1. *Hand Feeding***

Pemberian pakan dengan cara ini memiliki beberapa kelebihan, antara lain jumlah makanan yang diberikan lebih terkontrol, terbatas, dan tidak banyak terbuang. Akan tetapi, cara ini juga memiliki kerugian, antara lain membutuhkan banyak tenaga dan waktu. Yang lebih penting untuk diingat adalah kelinci termasuk hewan malam yang aktivitasnya dilakukan pada malam hari, sehingga ransum banyak diberikan pada sore hari.

###### **2. *Self Feeding***

*Self feeding* dilakukan dengan cara makanan diberikan sekaligus sehingga kelinci dapat makan secara bebas. Cara ini biasa dilakukan pada kelinci yang bertujuan untuk digemukkan. Kelebihan pemberian makan dengan cara ini adalah tidak membutuhkan tenaga dan waktu yang banyak. Sedangkan kerugiannya adalah ransum yang dikonsumsi dan dikencingi tidak dapat dikonsumsi lagi (Alex S, 2014).

## 2.2 Landasan Teori

### 2.2.1 Telegram

Telegram adalah aplikasi android yang berupa pesan *chatting* seperti, Whatsapp, dan Line. Telegram Messenger telah menggunakan protocol MTProto yang sudah teruji dengan tingkat keamanannya karena proses enkripsi *end to end* yang digunakan.

Pada alat “Pemberi Pakan Kelinci Terjadwal dan Monitoring Stok Pakan Melalui Telegram” ini, Telegram mempunyai fungsi yang sangat besar, yaitu dengan adanya aplikasi Telegram *messenger*, alat ini dapat dipantau dengan menggunakan bot pada Telegram. Bot Telegram adalah sebuah bot atau robot yang diprogram dengan berbagai perintah untuk menjalankan serangkaian instruksi yang diberikan oleh pengguna (Hakim, 2021).

### 2.2.2 Bot Telegram

Telegram bot adalah sebuah bot atau robot yang diprogram dengan berbagai perintah untuk menjaankan serangkaian intruksi yang diberikan oleh pengguna melalui chat dengan platform Telegram. Bot ini hanyalah sebuah akun Telegram yang dioperasikan oleh perangkat lunak yang memiliki fitur AI (Anonymous1, n.d.).



Gambar 2.1. Logo BotFather

Di Telegram, semua bot di "lahir"-kan (dibuat/didaftarkan) dari seorang bapak. Bapak Bot dalam bahasa Inggris adalah BotFather. Jadi jika

ingin membuat bot, pertama kali yang harus dilakukan adalah mendaftarkan secara administratif ke BotFather.

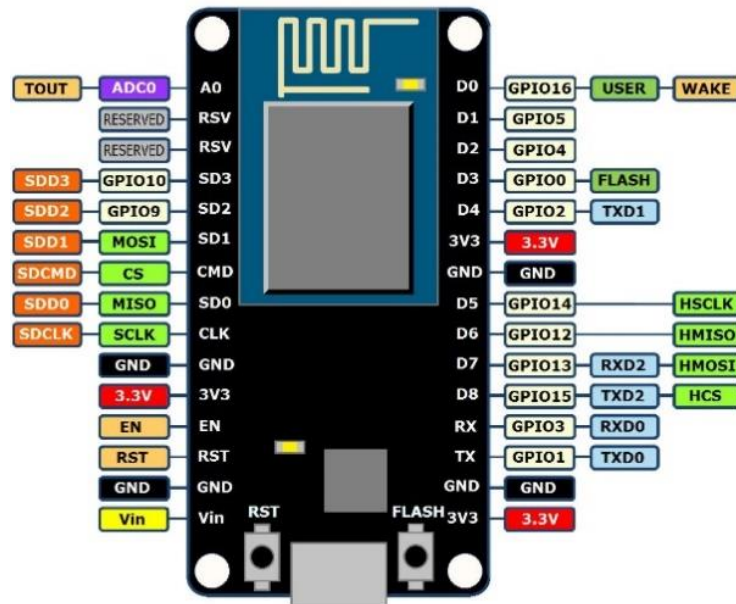
Di situ akan ditanya (memasukkan) terlebih dahulu 2 hal yang paling dasar, yakni nama dan username bot. Informasi lainnya (deskripsi, foto, pengaturan, dll) bisa menyusul. Jadi jelas fungsi utama bapak bot di sini adalah untuk mendaftarkan bot pertama kali.

Fungsi kedua adalah mengatur bot yang sudah dibuat. Yakni, hak akses, memberi keterangan deskripsi, mengatur foto profil bot, mengaktifkan/non aktifkan mode inline, mengatur set perintah, dlsb (Hasanudin, 2020).

Telegram memberikan beberapa identitas untuk para penggunanya. Identitas tersebut berupa ID dan username (Anonymous2, 2021). Untuk mendapatkan ID tersebut bisa menggunakan bot yang menyediakan fitur tersebut. Contohnya IDBot.

### **2.2.3 NodeMCU ESP8266**

NodeMCU adalah *platform* yang bersifat *open source* dan pengembangan kit menggunakan bahasa Lua untuk membantu dalam pembuatan *prototype* produk atau bisa dengan memakai *sketch* dengan arduino IDE. Pengembangan kit ini didasarkan pada modul ESP8266, yang mengintegrasikan GPIO, PWM (*Pulse Width Modulation*), IIC, 1-Wire dan ADC (*Analog to Digital Converter*) semua dalam satu board. GPIO NodeMCU ESP8266 seperti Gambar 2.2.



Gambar 2.2. NodeMCU ESP8266

GPIO (*General Purpose Input Output*) adalah pin *generic* pada sirkuit terpadu (*chip*) dapat dikontrol dan deprogram. GPIO bias *full control* lewat jaringan wifi. Di bawah ini merupakan tabel penamaan pin NodeMCU untuk Arduino IDE (Nugroho, 2020).

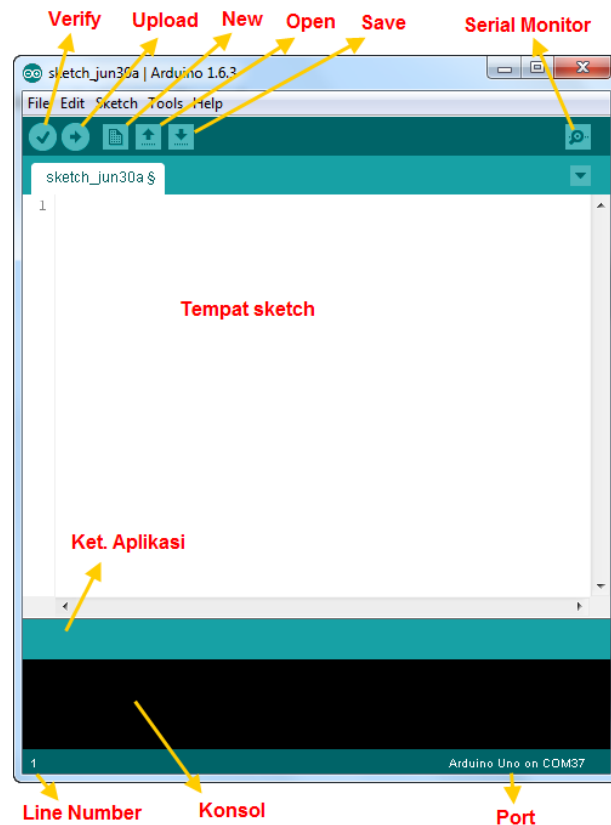
Tabel 2.1. Mapping Pin NodeMCU untuk Arduino

<u>NodeMCU</u>		<u>Arduino</u>	
IO Number	Pin Name	Pin Name	IO Number
GPIO 0	D3	D0	16
GPIO 1	D10	D1	5
GPIO 2	D4	D2	4
GPIO 3	D9	D3	0
GPIO 4	D2	D4	2
GPIO 5	D1	D5	14
GPIO 9	SD2	D6	12
GPIO 10	SD3	D7	13
GPIO 12	D6	D8	15
GPIO 13	D7	D9	3
GPIO 14	D5	D10	1
GPIO 15	D8	SD2	9
GPIO 16	D0	SD3	10

#### 2.2.4 Software Arduino IDE

Untuk memprogram *board* ESP8266, kita butuh aplikasi IDE (*Integrated Development Environment*) bawaan dari Arduino. Aplikasi ini

berguna untuk membuat, membuka, dan mengedit *source code* Arduino. *Sketch* merupakan *source code* yang berisi logika dan algoritma yang akan diupload ke dalam IC mikrokontroler (ESP8266).



Gambar 2.3. Interface Arduino IDE

*Interface* Arduino IDE tampak seperti gambar 2.2.3, bagian-bagian Arduino IDE terdiri dari :

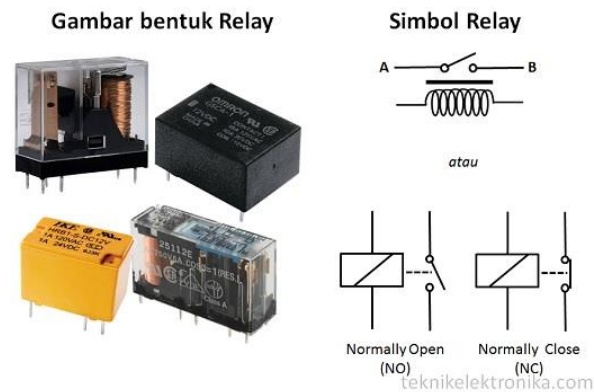
- **Verify**: Proses *verify* mengubah *sketch* ke *binary code* untuk diupload ke Arduino.
- **Upload**: tombol ini berfungsi untuk mengupload *sketch* ke *board* mikrokontroler. Walaupun kita tidak mengklik tombol *verify*, maka *sketch* akan di-*compile*, kemudian langsung di-*upload* ke *board*.
- **New** : Membuka jendela dan membuat *sketch* baru.
- **Open** : Membuka *sketch* yang sudah pernah dibuat. *Sketch* yang dibuat dengan Arduino IDE akan disimpan dengan ekstensi file *.ino*
- **Save** : menyimpan *sketch*, tapi tidak disertai meng-*compile*.

- Serial Monitor : Membuka *interface* untuk komunikasi serial.
- Keterangan Aplikasi : pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal "*Compiling*" dan "*Done Uploading*" ketika kita meng-*compile* dan meng-*upload sketch* ke *board* Arduino.
- Konsol : Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang *sketch* akan muncul pada bagian ini. Misal, ketika aplikasi meng-*compile* atau ketika ada kesalahan pada *sketch* yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.
- Baris Sketch: bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada *sketch*.
- Informasi Port : bagian ini menginformasikan port yang dipakai oleh *board* Arduino (Santoso, 2015).

Pada alat Pemberi Pakan Kelinci Terjadwal dan Monitoring Stok Pakan Melalui Telegram, juga menggunakan *library* ArduinoJson dan CT.Bot untuk menjalankan NodeMCU dan aplikasi Telegram *messenger*, dimana ArduinoJson merupakan *library* untuk mengaktifkan NodeMCU, dan CT.Bot merupakan *library* yang berfungsi untuk koneksi ke aplikasi Telegram, pastikan kedua *library* sudah terinstal dalam *software* Arduino IDE.

### 2.2.5 Relay

Relay adalah saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (*Coil*) dan Mekanikal (seperangkat kontak saklar/*switch*). Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan relay yang menggunakan elektromagnet 5V dan 50 mA mampu menggerakkan Armature Relay (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A. Dibawah ini adalah gambar bentuk relay dan simbol relay yang sering ditemukan di rangkaian elektronika.



Gambar 2.4. Gambar dan Simbol Relay

Kontak Poin (*Contact Point*) Relay terdiri dari 2 jenis yaitu :

1. *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *close* (tertutup)
2. *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *open* (terbuka) (Kho, 2020).

### 2.2.6 RTC DS3231

Serial RTC DS3231 merupakan *low-power* dan *full binary-coded-decimal* (BCD). Data dan alamat ditransfer berurutan secara serial melalui dua kabel dan bidirectional bus. *Clock/Calendar* menyediakan detik, menit, jam, hari, tanggal, bulan dan informasi tahun. Akhir dari tanggal dan bulan secara otomatis akan disesuaikan selama sebulan paling sedikit 31 hari yang mencakup koreksi untuk tahun kabisat. Jam beroperasi dalam format 12 jam atau 24 jam dengan AM/PM indikator (Kusumawati & Wiryanto, 2018).

RTC DS3231 adalah salah satu jenis modul yang berfungsi sebagai pewaktu digital, modul ini juga memiliki fungsi pengukur suhu yang dikemas menjadi satu (Rienaldhy & Ridho, 2021).



Gambar 2.5. RTC DS3231

Keterangan :

- Vcc = 5V,
- GND = Ground,
- SDA = Serial Data(12c),
- SCL = Serial Clock(12c).

### 2.2.7 Motor Servo

Motor servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem *closed feedback* yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (axis) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Contoh motor servo dapat dilihat pada Gambar 2.6.



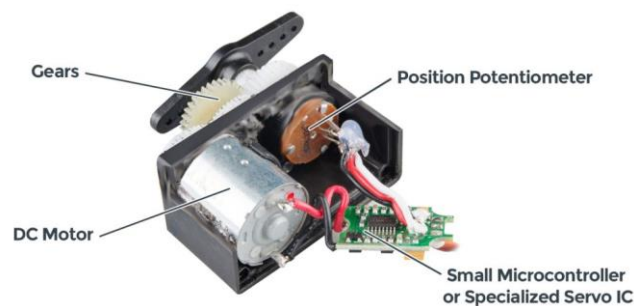
Gambar 2.6. Servo

Motor servo disusun dari sebuah motor DC, *gearbox*, variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor



servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo.

Motor servo pada dasarnya terdiri dari motor DC, rangkaian *gearbox*, rangkaian kontrol dan potensiometer. Bagian-bagiannya dengan jelas dapat dilihat pada Gambar 2.7 berikut:



Gambar 2.7. Komponen Penyusun Motor Servo

Pada motor servo kontrol, motor terpasang oleh roda gigi ke roda kontrol. Saat motor berputar, resistansi potensiometer akan berubah, sehingga rangkaian kontrol dapat dengan tepat mengatur seberapa banyak gerakan yang ada dan ke arah mana. Ketika poros motor berada pada posisi yang diinginkan, daya yang disuplai ke motor dihentikan. Jika tidak, motor diputar ke arah yang sesuai. Posisi yang diinginkan dikirim melalui pulsa listrik melalui kabel sinyal. Kecepatan motor sebanding dengan perbedaan antara posisi aktual dan posisi yang diinginkan. Jadi jika motor berada di dekat posisi yang diinginkan, ia akan berputar perlahan, jika tidak maka akan berputar cepat, ini disebut sebagai kontrol proporsional.

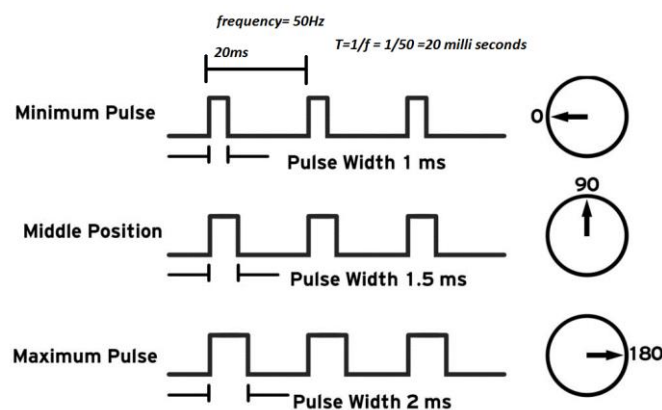
Prinsip Kerja Motor Servo yaitu Servo dikontrol dengan mengirimkan pulsa listrik dengan lebar variabel, atau biasa disebut Pulse Width Modulation (PWM) melalui kabel kontrol. Ada pulsa minimum, pulsa maksimum, dan tingkat pengulangan. Motor servo biasanya hanya bisa berputar  $90^\circ$  ke arah manapun dengan total gerakan  $180^\circ$ . Posisi netral motor didefinisikan sebagai posisi di mana servo memiliki jumlah putaran

potensial yang sama di kedua arah searah jarum jam atau berlawanan arah jarum jam (Anonymous5, 2022).

Untuk tetap mempertahankan posisinya, rangkaian kontrol memerlukan sinyal PWM (Pulse Width Modulation). Lebar sinyal ini diatur diantara 1ms hingga 2ms (milidetik). Motor akan berputar dari titik  $0^\circ$  hingga maksimal ( $180^\circ$  atau  $360^\circ$ , tergantung tipenya) jika diberikan sinyal pada rentang waktu tersebut. Sinyal PWM ini harus terus diberikan setiap 20ms (Anonymous4, 2021).

Contohnya, pulse 1,5ms akan membuat motor berputar ke posisi  $90^\circ$ . Apabila Lebih pendek dari 1.5ms, maka motor akan bergerak berlawanan ke arah jarum jam menuju posisi  $0^\circ$ , dan apabila lebih dari 1.5ms akan mengubah putaran servo ke arah searah jarum jam menuju posisi  $180^\circ$ .

Perhatikan Gambar 2.8 di bawah ini. Pada lebar pulsa 1ms, motor akan tetap pada posisi  $0^\circ$ . Saat lebar pulsa diubah menjadi 1,5ms motor akan berputar  $90^\circ$ , dan apabila diberi sinyal 2ms maka putarannya menjadi  $180^\circ$ .



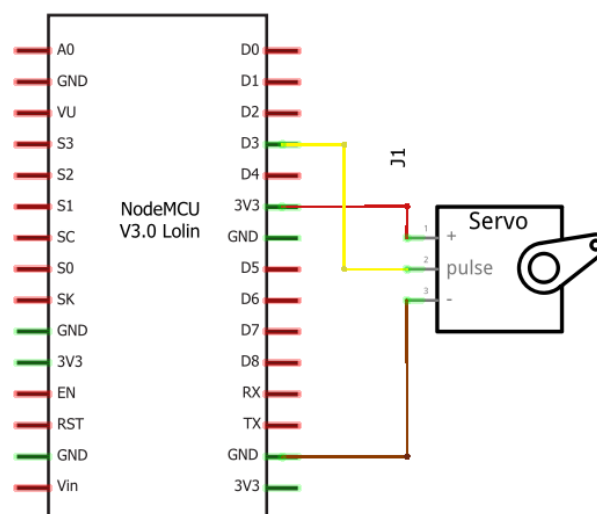
Gambar 2.8. Pemberian *Pulse* pada Motor Sevo

Ketika servo ini diperintahkan untuk bergerak, maka dia akan bergerak dan diam diposisi tersebut. Jika servo dipaksa untuk bergerak dari posisi diam, maka servo akan menolak bergerak diluar dari posisi tersebut.

Jumlah gaya maksimum yang dapat diberikan oleh servo disebut tingkat torsi servo. Servo tidak akan diam berada pada posisi tersebut selamanya, sehingga pulsa posisi harus diulang untuk menginstruksikan servo untuk tetap dalam posisi (Anonymous5, 2022).

Pada Gambar 2.9 berikut ditunjukkan jalur kelistrikan dimana sinyal PWM dapat mengoperasikan sesuai deskripsi yang telah diuraikan.

Kabel coklat pada servo terhubung dengan pin negatif pada NodeMCU. Lalu kabel merah yang merupakan kabel power, terhubung pada pin positif pada NodeMCU. Dan kabel yang ketiga berwarna kuning yang merupakan kabel pulsa/data, terhubung pada pin D3 atau pin digital pada NodeMCU.



Gambar 2.9. Skema Motor Servo

### 2.2.8 Sensor Ultrasonik

Sensor ultrasonik HCSR-04 dapat digunakan untuk mengukur jarak antara objek dan sensor. Cara kerja sensor ultrasonik dengan mengirimkan gelombang tertentu, kemudian menghitung waktu ketika diterima kembali oleh sensor. Sensor ini sudah tersedia modul *transmitter* dan *receiver* gelombang ultrasonik. Berikut ini spesifikasi dari sensor HC-SR04.

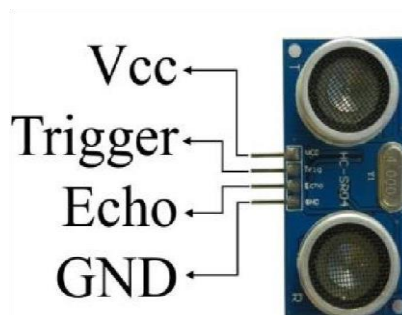
Tabel 2.2. Spesifikasi Sensor HC-SR04

Power Supply	+5V DC
Arus daya	15mA
Sudut efektif	15
Pembacaan jarak	2cm – 400cm
Pengukuran Sudut	30

Tabel 2.3. Spesifikasi Pin pada Sensor HC-SR04

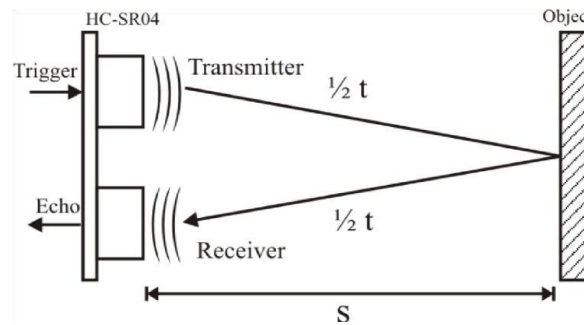
Nama Pin	Keterangan
VCC	Sumber tenaga (5V)
Trig	Pemicu sinyal sonar dari sensor
Echo	Penangkap sinyal sonar dari sensor
GND	Ground

Konfigurasi pin dan tampilan sensor HC-SR04 diperlihatkan pada gambar di bawah ini



Gambar 2.10. Konfigurasi Pin dan Tampilan Sensor Ultrasonik HC-SR04

HC-SR04 memiliki 2 komponen utama sebagai penyusunnya yaitu *ultrasonic transmitter* dan *ultrasonic receiver*. Fungsi dari *ultrasonic transmitter* adalah memancarkan gelombang ultrasonik dengan frekuensi 40 KHz kemudian *ultrasonic receiver* menangkap hasil pantulan gelombang ultrasonik yang mengenai suatu objek. Waktu tempuh gelombang ultrasonik dari pemancar hingga sampai ke penerima sebanding dengan 2 kali jarak antara sensor dan bidang pantul seperti yang diperlihatkan pada gambar berikut:



Gambar 2.11. Prinsip Kerja Sensor HC-SR04

Prinsip pengukuran jarak menggunakan sensor ultrasonik HC-SR04 adalah ketika pulsa trigger diberikan pada sensor, *transmitter* akan mulai memancarkan gelombang ultrasonik, pada saat yang sama sensor akan menghasilkan output TTL transisi naik menandakan sensor mulai menghitung waktu pengukuran, setelah *receiver* menerima pantulan yang dihasilkan oleh suatu objek maka pengukuran waktu akan dihentikan dengan menghasilkan output TTL transisi turun. Jika waktu pengukuran adalah  $t$  dan kecepatan suara adalah  $340 \text{ m/s}$ , maka jarak antara sensor dengan objek dihitung dengan rumus :

$$s = \frac{340 \left( \frac{100}{1000000} \right) \cdot t}{2}$$

$$s = \frac{0.034 \cdot t}{2}$$

$s$  = Jarak antara sensor dengan objek (cm)

$t$  = Waktu tempuh gelombang ultrasonik dari *transmitter* ke *receiver* (s)

Diketahui bahwa kecepatan suara adalah  $v = 340 \text{ m/s}$  atau  $0,034 \text{ cm}/\mu\text{s}$  karena kita ingin jaraknya dalam bentuk satuan cm. Untuk menghitung jarak, gunakan persamaan  $s = v \cdot t$ . Karena  $v = 0,034 \text{ cm}/\mu\text{s}$ . Maka  $s = 0,034 \cdot t$ .

Tidak hanya sampai disitu, karena waktu tempuh gelombang suara adalah dua kali yaitu saat pertama dikeluarkan dan setelah memantul dari benda kembali ke sensor maka persamaan tadi menjadi  $s = 0,034 \cdot t/2$  (Tedy Tri Saputro, 2018).

Kecepatan suara ( $v$ ) = 340 m/s atau 0,034 cm/ $\mu$ s.

Rumus jarak :

$$s = v * t$$

$$s = 0,034 * t$$

Maka didapat rumus jarak pada sensor ultrasonik HC-SR04 :

$$s = 0,034 * \frac{t}{2}$$

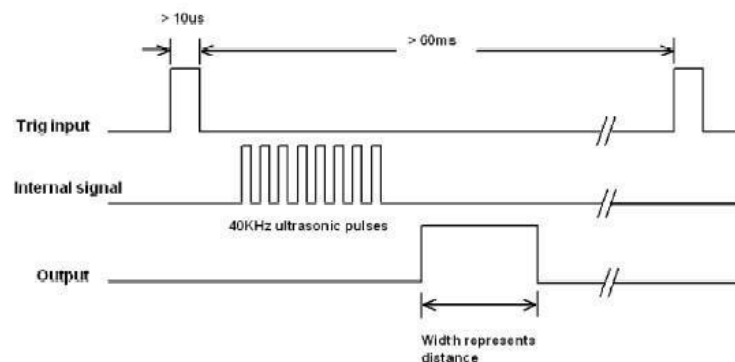
$s$  = Jarak antara sensor dengan objek (cm)

$t$  = Waktu tempuh gelombang ultrasonik dari *transmitter* ke *receiver* ( $\mu$ s)

2 = dua kali gelombang suara yang pertama dikeluarkan dan setelah itu memantulkan dari benda kembali ke sensor.

Prinsip pengoperasian sensor ultrasonik HC-SR04 adalah sebagai berikut: awali dengan memberikan pulsa *Low* (0) ketika modul mulai dioperasikan, kemudian berikan pulsa *High* (1) pada trigger selama 10  $\mu$ s sehingga modul mulai memancarkan 8 gelombang kotak dengan frekuensi 40 KHz, tunggu hingga transisi naik terjadi pada output dan mulai perhitungan waktu hingga transisi turun terjadi, setelah itu gunakan persamaan rumus di atas untuk mengukur jarak antara sensor dengan objek (Nugroho, 2020).

*Timing* diagram diperlihatkan pada gambar berikut.



Gambar 2.12. Timing Diagram Pengoperasian Sensor Ultrasonik HC-SR04

### 2.2.9 Teori Ralat

Tujuan pengukuran adalah untuk mengetahui harga/nilai antara besaran yang diukur dengan besaran yang dianggap tolok ukurnya. Dalam kenyataannya, nilai pembanding yang sesungguhnya tidak pernah diketahui sehingga hasil pengukuran yang benar tidak pernah diketahui.

Setiap kali melakukan pengukuran yang diulang-ulang dengan teliti, hasilnya hampir selalu berbeda meskipun selisihnya sangat kecil. Karenanya dalam proses pengukuran selalu terdapat kesalahan atau ralat ("error"). Usaha yang harus dilakukan dalam setiap pengukuran adalah memperoleh kesalahan tersebut sekecil mungkin.

Jika pengamatan/pengukuran dilakukan berkali-kali pada besaran yang diukur secara langsung, hasilnya berbeda-beda, misalnya hasil pengamatan/pengukuran yang dilakukan sebanyak  $k$  kali dengan hasil tiap kali  $x_i$  :  $x_1$  ;  $x_2$  ;  $x_3$  . . . . .  $x_k$ , dimana  $x_i$  yang besarnya  $x_1$  ;  $x_2$  ;  $x_3$  . . . . .  $x_k$ , dinamakan nilai terukur yang merupakan nilai atau harga yang mungkin. Nilai terbaik dari nilai-nilai terukur adalah nilai rata-ratanya yang juga merupakan nilai yang paling mungkin, jadi nilai terbaiknya ( $\bar{x}$ ) yaitu :

$$\bar{x} = \frac{\sum_{i=1}^k x_i}{k} = \frac{x_1 + x_2 + \dots + x_k}{k} \dots \dots \dots (1)$$

Selisih atau penyimpangan antara nilai terukur dengan nilai rata-rata disebut deviasi dengan lambang  $s$ , jadi :

$$\delta_x = x_i - \bar{x} \quad (2)$$

Deviasi seperti yang dituliskan pada persamaan (2), merupakan penyimpangan terhadap nilai terbaik dari nilai terukur yang bersangkutan ( $x_i$ ). Untuk menentukan nilai pengamatan yang mungkin, ditentukan nilai terbaik ( $\bar{x}$ ) dengan penyimpangan yang disebut "deviasi standard". Deviasi standard ini didefinisikan sebagai akar rata-rata kuadrat deviasinya, dan

untuk pengamatan di Lab. Fisika Dasar umumnya besaran terukur tunggal digunakan rumus :

$$\text{Standard deviasi} = s_{\bar{x}} = \sqrt{\frac{\sum_{i=1}^k (\delta_{x_i})^2}{k(k-1)}} = \sqrt{\frac{\sum_{i=1}^k (x_i - \bar{x})^2}{k(k-1)}} \quad (3)$$

Sedangkan deviasi standard relatifnya dapat ditulis :

$$s_{xr} = \frac{s_x}{\bar{x}} \quad \text{atau} \quad s_{xr} = \frac{s_x}{\bar{x}} \times 100\% \quad (4)$$

Dengan demikian maka harga atau nilai suatu pengukuran/pengamatan dapat ditulis besaran-besaran  $x$  yang benar adalah  $\bar{x} \pm s_x$ , jadi

$$x = \bar{x} \pm s_x \quad (5)$$

Dalam menyatakan nilai suatu pengukuran, seringkali dinyatakan dengan kesaksamaan atau kecermatan, yaitu  $1 - s_{xr}$  atau  $100\% - s_{xr} \%$ . Kesaksamaan atau kecermatan dapat dianggap sebagai jaminan akan kebenaran hasil pengamatan (Aminudin, 2014).

### 2.2.10 Konfigurasi Arduino IDE

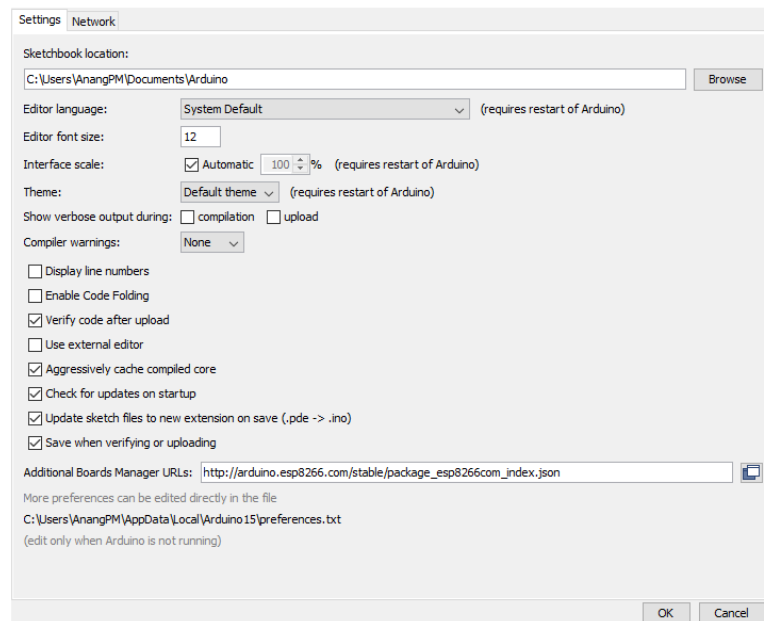
Konfigurasi ini dilakukan karena pada Arduino IDE standar tidak terdapat *library* untuk ESP8266 sehingga perlu melakukan instalasi *addon* ESP8266 Library.

#### 1. Instalasi addon ESP8266 dengan Arduino Board Manager

Untuk buka Arduino IDE kemudian masuk ke menu preferences kemudian di bagian bawah jendela masukkan url berikut:

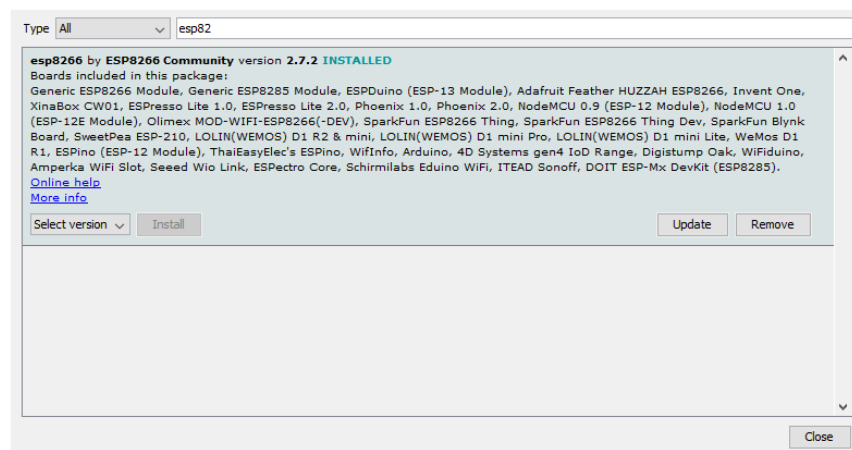
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)





Gambar 2.13. Menu Preferences Arduino IDE

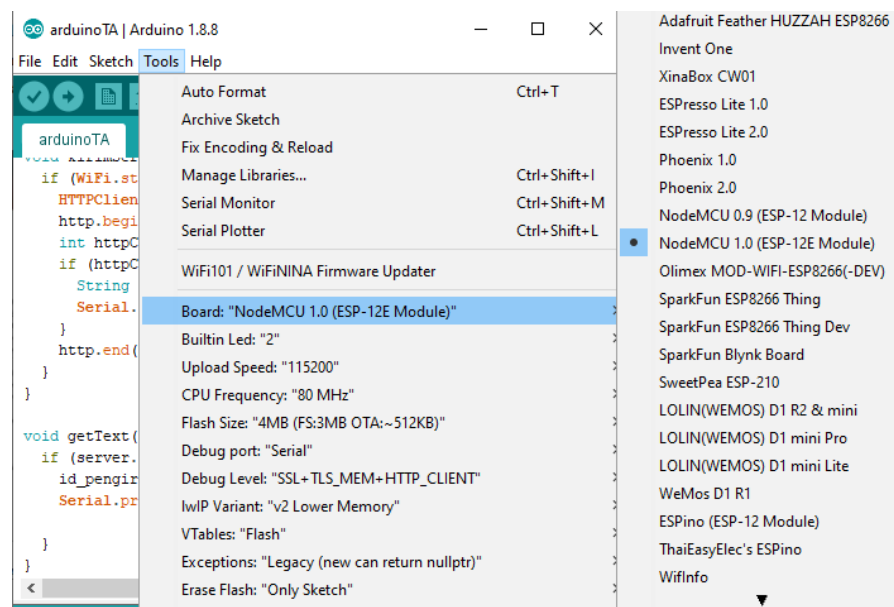
2. Setelah menambahkan URL pada Additional Board Manager kemudian masuk ke menu Board Manager (Tools > Boards > Boards Manager). Kemudian cari ESP8266 pada Filter untuk memudahkan pencarian. Setelah itu klik Install.



Gambar 2.14. Menu Board Manager Arduino IDE

3. Memilih board ESP8266

Pilih board yang akan digunakan, pada proyek akhir ini jenis ESP8266 yang digunakan pada IDE adalah NodeMCU 1.0 (ESP-12E Module) (Nugroho, 2020). Seperti pada Gambar 2.15.



Gambar 2.15. Menu Setting Board Arduino IDE

### 2.3 Tinjauan Pustaka

Tinjauan pustaka merupakan acuan utama dalam beberapa studi yang pernah dilakukan dan berkaitan dengan penelitian. Terdapat beberapa penelitian yang digunakan sebagai rujukan dalam proyek akhir ini.

Tugas akhir Repan Pebri Nugroho dengan judul “Sistem Monitoring Pemberi Pakan Sapi Otomatis Secara Realtime Berbasis IoT”, pada tugas akhir ini yang dijadikan sebagai rujukan adalah menggunakan mikrokontroler NodeMCU ESP8266 serta sensor sebagai inputan yaitu sensor ultrasonik.

Tugas akhir Ahmad Fiveth Nazidul Faroh dengan judul “Pembersih Sampah pada Sungai Berbasis Internet Of Things”, pada tugas akhir ini yang dijadikan sebagai rujukan adalah menggunakan sensor ultrasonik sebagai inputan guna memantau keadaan tumpukan sampah, dan menggunakan aplikasi Telegram sebagai media notifikasinya.

Jurnal Penjadwalan Pakan Ikan Koi Otomatis Pada Kolam Menggunakan RTC DS3231 (Andriawan, 2018), yang dijadikan rujukan pada jurnal ini adalah menggunakan modul RTC DS3231 yang berfungsi untuk mengatur jadwal pemberian pakan.

Tugas Akhir Deni Rhamdani Sutiyono dengan judul “Rancang Bangun Sistem Kebersihan Kandang dan Pemberian Pakan pada Budidaya Kelinci Pedaging di Peternakan Asep Rabbit Ranch Project Lembang”. Pada tugas akhir ini yang dijadikan rujukan adalah objek yang digunakan yaitu kelinci dan penggunaan motor servo sebagai buka tutup katup pakan.