

MODUL 1

PENGENALAN MIKROKONTROLER

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umunya dapat menyimpan program didalamnya. Mikrokontroler umumnya terdiri dari CPU (Central Processing Unit), memori, I/O tertentu dan unit pendukung seperti Analog-to-Digital Converter (ADC) yang sudah terintegrasi di dalamnya. Kelebihan utama dari mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas.

Sistem yang menggunakan mikrokontroler sering disebut sebagai embedded system atau dedicated system. Embedded system adalah sistem pengendali yang tertanam pada suatu produk, sedangkan dedicated system adalah sistem pengendali yang dimaksudkan hanya untuk suatu fungsi tertentu. Sebagai contoh printer adalah suatu embedded system karena di dalamnya terdapat mikrokontroler sebagai pengendali dan juga dedicated system karena fungsi pengendali tersebut berfungsi hanya untuk menerima data dan mencetaknya. Hal ini berbeda dengan suatu PC yang dapat digunakan untuk berbagai macam keperluan, sehingga mikroprosesor pada PC sering disebut sebagai general purpose microprocessor (mikroprosesor serba guna). Pada PC berbagai macam software yang disimpan pada media penyimpanan dapat dijalankan, tidak seperti mikrokontroler hanya terdapat satu software aplikasi.

Beberapa karakteristik mikrokontroler antara lain sebagai berikut:

1. Memiliki program khusus yang disimpan didalam memori, program pada mikrokontroler relatif lebih kecil dibandingkan dengan program-program pada PC.
2. Murah, karena komponen-komponennya tidak dirancang untuk menghasilkan kemampuan komputasi yang tinggi.
3. Unit I/O yang sederhana , misalnya keypad, LCD dan LED.

4. Konsumsi daya kecil.
5. Rangkaian sederhana dan kompak.
6. Lebih tahan terhadap kondisi lingkungan ekstrem, misalnya temperature, tekanan dan kelembaban yang tinggi

A Mikroprosesor VS Mikrokontroler

Tabel 1.1. Perbandingan Mikroprosesor dengan Mikrokontroler

Mikroposeor	Mikrokontroler
Suatu rangkaian digital yang terdiri atas 3 bagian utama, yaitu: <ul style="list-style-type: none"> - ALU (<i>Aritmetic and Logic</i>), - Register(untuk penyimpanan data sementara), - Pengendali dan Pewaktuan (<i>control and timing</i>) yang dirancang sedemikian rupa dengan dimensi yang sangat kecil dan dibuat dalam satu chip. 	Suatu mikroprosesor yang telah dikombinasikan dengan komponen pendukung memori dan fitur I/O maupun clocking dan saru chip IC.

Perbedaan mikrokontroler dengan mikroprosesor secara umum adalah mikrokontroler dapat bekerja sendiri tanpa tambahan IC lain, karena sudah adanya memori dan fitur I/O, sedangkan mikroprosesor tidak dapat bekerja sendiri.

B Penggunaan Mikrokontroler

Penggunaan mikrokontroler antara lain terdapat pada bidang-bidang berikut ini.

1. Otomotif : Engine Control Unit, Air Bag, fuel control, Antilock Braking System, sistem pengaman alarm, transmisi otomatis, hiburan, pengkondisi udara, speedometer dan odometer, navigasi, suspensi aktif
2. Perlengkapan rumah tangga dan perkantoran : sistem pengaman alarm, remote control, mesin cuci, microwave, pengkondisi udara, timbangan digital, mesin foto kopi, printer, mouse.
3. Pengendali peralatan di industri.
4. Robotika.

C ARDUINO

Arduino merupakan pengendali mikro single-board (papan-tunggal mikrokontroler) yang bersifat open-source, yang dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Platform Arduino saat ini menjadi sangat populer karena penggunaannya yang mudah serta penulisan program yang tidak rumit seperti kebanyakan mikrokontroler. Dalam penggunaannya Arduino tidak membutuhkan perangkat keras terpisah seperti downloader untuk memuat atau meng-upload program baru ke dalam mikrokontroler, cukup menggunakan kabel USB untuk menggunakan Arduino. Selain itu, Arduino IDE menggunakan bahasa pemrograman C++ sehingga lebih mudah untuk dipelajari. Dengan banyaknya kemudahan, Arduino menjadi bentuk standari dari fungsi mikrokontroler yang mudah diakses.

Arduino dirancang untuk digunakan oleh siapa saja dari berbagai profesi yang tertarik dalam menciptakan objek interaktif dan pengembangan lingkungan. Arduino juga dapat berinteraksi dengan komponen lain seperti tombol, LED, motor, speaker, GPS, kamera, internet, smartphone, hingga televisi. Fleksibilitas tersebut dihasilkan dari kombinasi hardware yang mudah didapat serta ketersediaan software Arduino yang gratis karena berbasis open-source. Kombinasi tersebut membuat Arduino mudah untuk dipelajari. Hal ini menyebabkan jumlah pengguna bertambah dan berbagai kontribusi dalam berbagai proyek berbasis Arduino meningkat.



Gambar 1.1 Arduino UNO

Arduino Uno merupakan versi terbaru berbasis mikrokontroler ATmega328 yang dibuat untuk menyempurnakan tipe sebelumnya, yaitu Duemilanove. Perbedaan dari Arduino tersebut adalah tidak menggunakan IC FTDI (Future Technology Devices International) USB to Serial sebagai driver USB-nya akan tetapi menggunakan mikrokontroler ATmega8U2 yang diprogram sebagai converter USB ke serial.

Board Arduino Uno memiliki spesifikasi hardware berupa:

- Microcontroller: ATmega328
- Tegangan Output: 5V
- Tegangan Input: 7-12V
- Digital I/O: 14 pin
- PWM: 6 channel
- Analog Input: 6 channel
- Memory: 32KB Flash EPROM (0.5KB digunakan oleh bootloader), 2KB SRAM dan 1KB EEPROM
- Frekuensi Clock: 16 MHz

Berikut ini merupakan beberapa produk board yang diproduksi oleh Arduino.

Tabel 1.2. Produk Board Arduino

No	Nama Board	No	Nama Board
1	Arduino Uno	14	Lilypad Arduino
2	Arduino Leonardo	15	Arduino NG(Nouva Generazione)
3	Arduino Due	16	Arduino Extreme
4	Arduino Yun	17	Arduino Diecimila
5	Arduino Tre	18	Arduino Mega
6	Arduino Micro	19	Arduino Mega 2560
7	Arduino Robot	20	Arduino Pro
8	Arduino Esplora	21	Arduino Pro Mini
9	Arduino Mega ADK	22	Arduino Ethernet
10	Arduino Fio	23	Arduino Duemilanove
11	Arduino USB	24	Arduino Shields
12	Arduino Nano	25	Arduino Single Side Serial
13	Arduino Serial	26	Arduino Mini

Berikut ini merupakan penjelasan dari beberapa komponen utama Arduino Uno:

1. Power Supply

Pada board Arduino terdapat 2 pilihan sumber tegangan yaitu port USB dan power supply eksternal. Power supply yang bersumber dari port USB digunakan dengan cara menghubungkan port USB di komputer atau laptop dengan Arduino. Sedangkan untuk sumber tegangan eksternal didapatkan dengan cara menghubungkan Arduino board dengan sumber tegangan DC. Tegangan yang direkomendasikan sebesar 7 V hingga 12 V. Apabila sumber tegangan kurang dari 7 V akan menyebabkan ketidakstabilan tegangan, sedangkan jika sumber tegangan lebih dari 12 V akan menyebabkan panas hingga kerusakan fatal pada board Arduino.

2. Input-Output

Port Arduino memiliki penamaan yang berbeda dari sistem minimum mikrokontroler. Sistem minimum ATmega8535 memiliki penamaan port berupa PORTA, PORTB, PORTC, dan PORTD, untuk akses per bit maka PORTA.0 sampai PORTA.7. Contoh lain dari penamaan port pada sistem minimum adalah pada AT89S51 berupa PORT0, PORT1, dan seterusnya. Sistem penamaan port pada Arduino merupakan urutan nomor port, mulai dari nomor 0, 1, dan seterusnya. Pada digital I/O digunakan nama pin 1, 2, hingga 13. Sedangkan penamaan untuk analog input menggunakan nama A0, A1, hingga A5.

Arduino uno memiliki 14 pin digital input-output. Secara umum fungsi pin tersebut sebagai port input-output biasa, namun dalam beberapa hal pin tersebut memiliki fungsi alternative. Sebagai contoh, pin2 dan 3 juga digunakan sebagai pin interupsi eksternal. Selain itu pin 5,6,9,10, dan 11 dapat digunakan untuk PWM(Pulse Width Modulation) yang sering digunakan untuk kendali motor DC maupun motor servo.

Berikut ini merupakan tabel dari fungsi pin pada Arduino Uno.

Tabel 1.3. Fungsi PIN Board Arduino

No Pin	Fungsi	Fungsi Alternatif
0	Digital I/O 0	RX (serial-receiver)
1	Digital I/O 1	TX (serial-transmitter)
2	Digital I/O 2	Interupsi eksternal
3	Digital I/O 3	Interupsi eksternal & PWM
4	Digital I/O 4	-
5	Digital I/O 5	PWM
6	Digital I/O 6	PWM
7	Digital I/O 7	-
8	Digital I/O 8	-
9	Digital I/O 9	PWM
10	Digital I/O 10	SPI-SS & PWM
11	Digital I/O 11	SPI-MOSI & PWM
12	Digital I/O 12	SPI-MISO
13	Digital I/O 13	SPI-SCK & LED

3. Analog Input

Arduino memiliki 6 pin analog input yang berfungsi membaca sinyal masukan analog (seperti sensor analog). Akan tetapi pin analog input juga dapat digunakan untuk keperluan I/O. Berikut ini merupakan tabel yang menunjukkan nomor dan fungsi dari pin input analog.

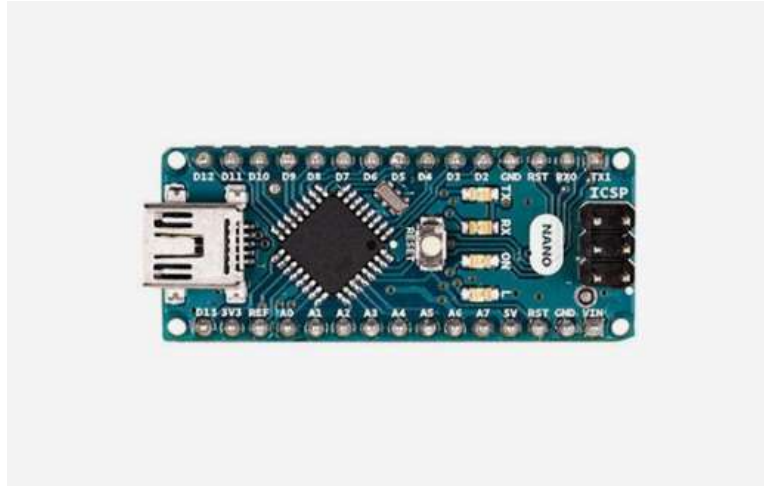
Tabel 1.4. Fungsi PIN Analog Board Arduino

No Pin	Fungsi	Fungsi Alternatif
A0	Analog Input 1	-
A1	Analog Input 2	-
A2	Analog Input 3	-
A3	Analog Input 4	-
A4	Analog Input 5	TWI-SDA
A5	Analog Input 6	TWI-SCL

Arduino memberikan berbagai kemudahan bagi penggunanya untuk membuat berbagai macam proyek berbasis mikrokontroler, seperti simulasi lampu, robot,

mengontrol motor dc, mengontrol motor stepper, pengatur suhu, display LCD, dan lain-lain.

D ARDUINO NANO



Gambar 1.2 Arduino NANO

Arduino Nano merupakan Arduino yang secara ukuran lebih kecil dibanding Arduino Uno. Nano menggunakan basis prosesor yang hampir sama dengan Uno yaitu Atmega 328 (Uno rev 3 menggunakan Atmega 328P). Nano menggunakan 1 tipe sumber daya yaitu 5v yang disuplai dari port micro USB atau pin VIN pada board. Ukurannya yang kecil membuat Nano cocok diintegrasikan pada breadboard mini dan rangkaian-rangkaian yang membutuhkan minimalitas dari segi ukuran.

Berikut perbandingan spesifikasi perangkat keras antara Arduino Uno dan Nano.

Tabel 1.1 Perbandingan Spesifikasi Arduino Uno dan Nano

Jenis	Family	SRAM/FLASH/EEPROM (KB)	Clock Speed	UART	PWM Pin	Digital Pin	Analog Pin
Uno	Atmega 328	2/32/1	16 Mhz	1	6	14	6
Nano	Atmega 328	2/32/1	16 Mhz	1	6	14	8

1. Power Supply

Arduino Nano menggunakan input daya listrik melalui konektor USB Mini (B), selain itu Nano juga bisa menerima daya listrik dari pin VIN dengan voltase sebesar 6-20V atau melalui pin 27 (5V) dengan voltase 5V teregulasi. Apabila terdapat lebih dari satu sumber daya listrik yang terhubung, maka Nano otomatis hanya akan menggunakan daya listrik dengan voltase terbesar.

2. Memory

Arduino Nano memiliki konfigurasi memori yang mirip dengan Uno, perbedaan hanya terdapat pada besar usable flash memory. Nano hanya memiliki 30 KB usable flash memory (2 KB dipakai oleh bootloader), sementara Uno memiliki 31,5 KB usable flash memory (hanya 0,5 KB yang dipakai oleh bootloader). Besar SRAM dan EEPROM dari kedua jenis Arduino ini adalah sama persis, yaitu 2 KB SRAM dan 1 KB EEPROM.

3. Input and Output

Setiap pin dari 14 digital pin pada Nano bisa digunakan sebagai pin Input atau Output dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Masing-masing pin beroperasi dengan tegangan sebesar 5 volt. Tiap pin mampu menyuplai arus listrik sebesar 40 mA.

Pin Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Kedua pin ini terhubung dengan dua pin pada chip FTDI USB-to-TTL Serial.

Pin interrupt eksternal : yaitu pin 2 dan 3. Kedua pin ini dapat diprogram untuk menjadi trigger interrupt pada nilai rendah (low value), ketika value meningkat atau menurun secara drastis, atau ketika terjadi perubahan spesifik (user defined) pada suatu value.

Pin-pin PWM : pin 3, 5, 6, 9, 10, dan 11. Berupa pin output PWM 8-bit dan diprogram dengan menggunakan fungsi `analogWrite()`.

Pin-pin SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin-pin ini mendukung komunikasi SPI (Serial Peripheral Interface).

Pin LED: pin 13. Nano memiliki LED terintegrasi yang juga tersambung ke pin 13. Jika pin diberi nilai High maka LED akan menyala, dan jika diberi nilai Low maka LED akan mati.

Nano memiliki 8 input analog yang masing-masing memiliki resolusi sebesar 10 bit (mampu menerima 1024 nilai input berbeda). Secara default pin menggunakan batas bawah 0V sampai ke 5V, walau dimungkinkan untuk mengubah batas atas tegangan dengan menggunakan fungsi `analogReference()`. Pin analog 6 dan 7 tidak dapat digunakan sebagai pin Digital.

Beberapa pin lain yang memiliki fungsi tambahan : Pin I2C: A4 (SDA) dan A5 (SCL). Mendukung komunikasi bus I2C (TWI) dengan memanfaatkan library yang bernama Wire.

Pin lain pada board Nano yaitu pin AREF, yang merupakan pin referensi voltase untuk input analog, AREF dikontrol dengan menggunakan fungsi `analogReference()`. Pin Reset digunakan untuk melakukan reset pada board Nano, biasanya dimanfaatkan untuk membuat tombol reset terpisah dari board.

4. Komunikasi

Nano memiliki beberapa cara untuk berkomunikasi dengan komputer/PC, board Arduino lain, atau mikrokontroler lain. Atmega328 memiliki fasilitas komunikasi serial UART TTL (5V) dengan menggunakan pin digital 0 (RX) dan 1 (TX). Selain itu terdapat juga chip FTDI FT232RL terintegrasi yang memungkinkan komunikasi serial melalui port USB. Selain itu driver FTDI memungkinkan port COM virtual-to-software pada komputer.

MODUL 2

PENGENALAN ARDUINO IDE

Arduino merupakan alat elektronik yang bersifat open-source dan mudah digunakan baik dari segi perangkat keras ataupun perangkat lunak. Arduino memiliki sebuah board yang digunakan sebagai input dan output. Input dapat berupa sensor, button, dan komunikasi data. Output dapat berupa motor, lampu led, dan komunikasi data.

Arduino dapat diprogram dengan menggunakan aplikasi Arduino Software (IDE) atau bisa juga secara Online menggunakan Arduino Web Editor.

Struktur Pemrograman Arduino

Secara umum struktur program pada Arduino berbasis pada Bahasa C yang terdiri atas 2 (dua) bagian besar yaitu Fungsi setup() dan Fungsi loop().

Fungsi Setup()

Fungsi setup() merupakan bagian program yang pertama kali dieksekusi ketika sketch berjalan. Inisialisasi yang ada pada fungsi setup() yaitu variabel, pin mode, library, serial. Setup() hanya dijalankan satu kali ketika program dijalankan atau ketika Arduino dalam kondisi reset.

```
int buttonPin = 3;

void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  // ...
}
```

Fungsi Loop()

Fungsi loop() dijalankan setelah fungsi setup() dan fungsi loop() merupakan program yang akan terus dijalankan selama alat dalam kondisi hidup. Inisialisasi pada fungsi loop() diantaranya yaitu inisialisasi nilai variabel, pengendalian program, dan proses.

```

const int buttonPin = 3;

// setup initializes serial and the button pin
void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

// loop checks the button pin each time,
// and will send serial if it is pressed
void loop()
{
  if (digitalRead(buttonPin) == HIGH)
    Serial.write('H');
  else
    Serial.write('L');

  delay(1000);
}

```

Komentar

Komentar diberikan untuk memberikan penjelasan program atau keterangan dari fungsi program. Penggunaan komentar didahului dengan tanda `/*` dan diakhiri dengan tanda `*/`, jika komentar lebih dari tiga baris maka setiap baris kecuali komentar pembuka `/*` dan komentar penutup `*/` didahului dengan tanda `*`.

```

/*
 * Program Arduino
 *
 * Contoh Arduino.
 * Komentar pada Arduino
 *
 * http://www.arduino.cc/
 */

```

Jenis komentar yang lain yaitu menggunakan komentar perbaris yang menggunakan tanda `//`.

```

int ledPin = 13;  //LED pada digital pin 13

```

Variabel

Variabel merupakan sebuah ruang penyimpanan data yang memiliki nama variabel, jenis tipe data, dan nilai variabel. Pembuatan variabel sering disebut deklarasi.

```

int pin = 13;

```

Seperti deklarasi variabel diatas yang memiliki nama variabel `pin`, dengan tipe data integer (`int`), dan memiliki nilai `13` dimana `13` merupakan nilai pin yang ada

pada Arduino Board. Untuk nilai variabel tidak harus nilai pin pada Arduino Board tetapi bisa nilai apapun dengan menyesuaikan kebutuhan data dan tipe data.

Fungsi

Fungsi adalah sekumpulan instruksi untuk mengerjakan suatu keperluan tertentu dengan hasil akhir pengembalian nilai dari keperluan tersebut.

```
void setup()
{
  pinMode(13, OUTPUT);
}
```

Nama dari fungsi diatas yaitu setup, badan fungsi dimulai dari tanda { dan diakhiri dengan }, sehingga program yang berada antara dua tanda tadi akan berjalan ketika fungsi dipanggil.

Fungsi pinMode()

Pada program fungsi diatas terdapat sebuah fungsi pinMode() yaitu fungsi untuk mendeklarasikan sebuah pin pada Arduino Board berdasarkan dua parameter. Parameter yang ada dalam fungsi pinMode(13, OUTPUT) ; yaitu 13 dimana nilai dari pin yang ada dalam Arduino Board bisa juga diisi dengan nama variabel yang sudah dideklarasikan dan OUTPUT yaitu jenis dari kegunaan pin tersebut bisa diisi dengan INPUT atau OUTPUT.

```
int ledPin = 13;
void setup()
{
  pinMode(ledPin, OUTPUT);
}
```

atau,

```
void setup()
{
  pinMode(13, OUTPUT)
}
```

Fungsi digitalWrite()

Fungsi digitalWrite() digunakan untuk memberikan nilai tegangan pada pin yang bersifat sebagai Output, jika tegangan diberi tegangan maka nilainya HIGH atau bisa dengan menggunakan nilai 1 sedangkan jika tidak diberi tegangan nilainya LOW atau bisa dengan menggunakan nilai 0.

```
digitalWrite(ledPin, HIGH);
```

Fungsi delay()

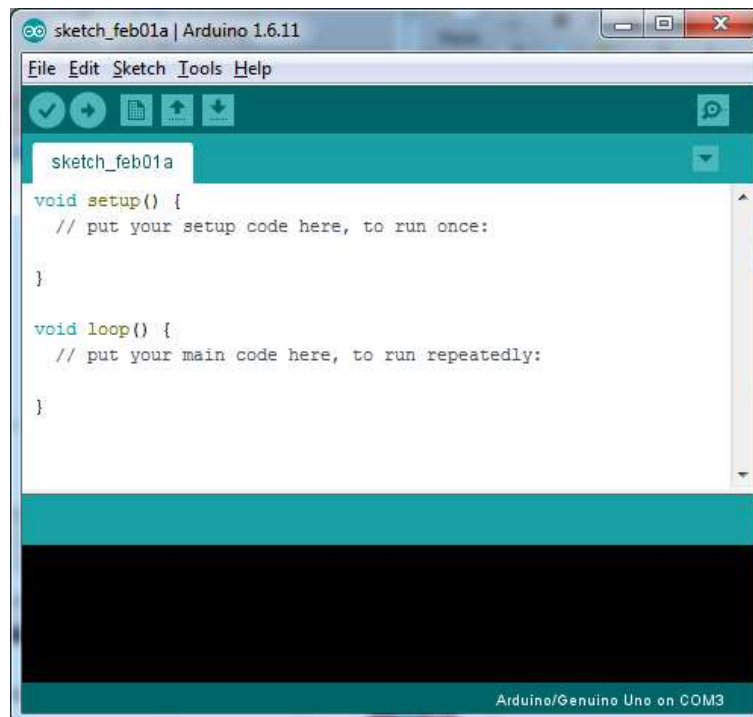
Fungsi delay() berfungsi untuk memberikan waktu jeda sebelum melanjutkan program selanjutnya. Delay yang diberikan satuannya yaitu miliseconds, jika dikonversikan ke second yaitu 1000 miliseconds.

```
delay(1000);
```

PENGUNAAN SKETCH PADA ARDUINO IDE






Untuk memberikan gambaran tentang penggunaan sketch pada Arduino IDE, silahkan mengikuti langkah-langkah berikut.

1. Buka aplikasi **Arduino**, akan muncul tampilan awal sebagai berikut.

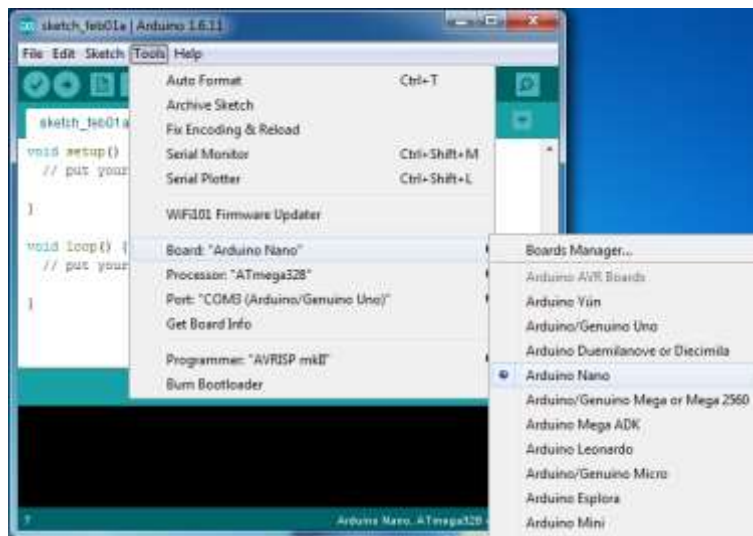


Gambar 2.1. Tampilan awal Arduino IDE

Pada tampilan diatas ada beberapa ikon yang akan sering digunakan ketika membuat program Arduino, diantaranya yaitu:

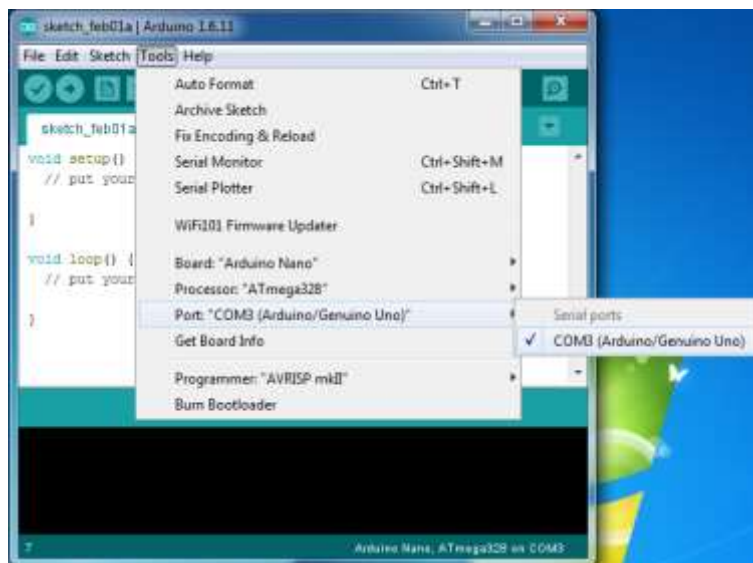
-  : Fungsi dari ikon disamping yaitu untuk compile sketch Arduino
-  : Fungsi dari ikon disamping yaitu untuk upload sketch ke Arduino
-  : Fungsi dari ikon disamping yaitu untuk membuat sketch baru
-  : Fungsi dari ikon disamping yaitu untuk membuka file sketch
-  : Fungsi dari ikon disamping yaitu untuk menyimpan file sketch

2. Lakukan pengaturan Boards pada sketch Arduino, sesuaikan dengan Arduino yang digunakan. Pilih **Tools** >> Klik **Board** >> Pilih Sub menu '**Arduino Nano**'.




Gambar 2.2. Tampilan menu pemilihan Board

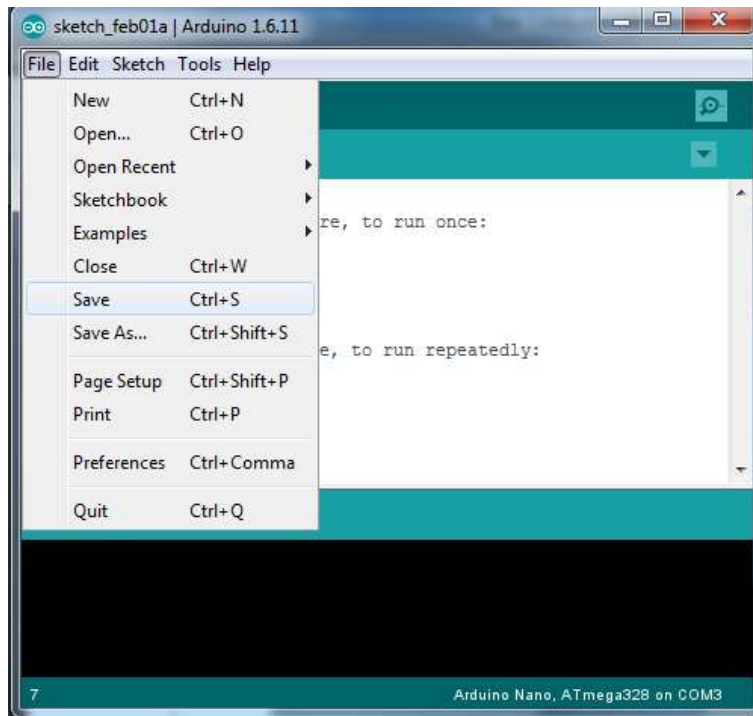
3. Kemudian atur Port yang digunakan pada Arduino, yaitu Port USB yang dipakai ke Arduino. Pilih **Tools** >> Klik **Port** >> **COMX**.



Gambar 2.3. Tampilan menu pemilihan Port

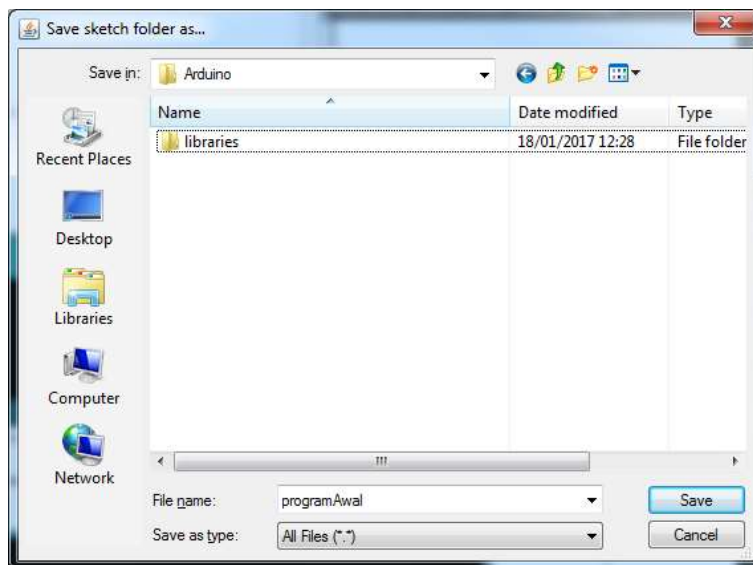
Pada port akan berbeda setiap komputer atau laptop, selain itu jenis Arduino juga berbeda.

4. Selanjutnya simpan sketch dengan menggunakan ikon  atau bisa juga dengan Pilih **File** >> Klik **Save**, selain itu dengan **Ctrl + S**.



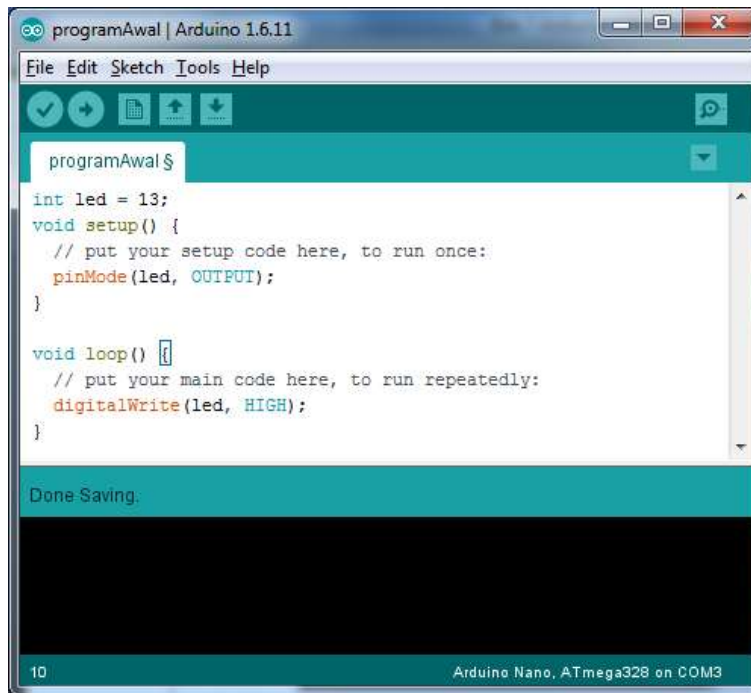
Gambar 2.4. Tampilan opsi Save

5. Setelah memilih Save maka akan muncul jendela Save, dan simpanlah pada drive yang disediakan sekaligus beri nama sketch tersebut. Seperti gambar berikut,



Gambar 2.5. Jendela Save file

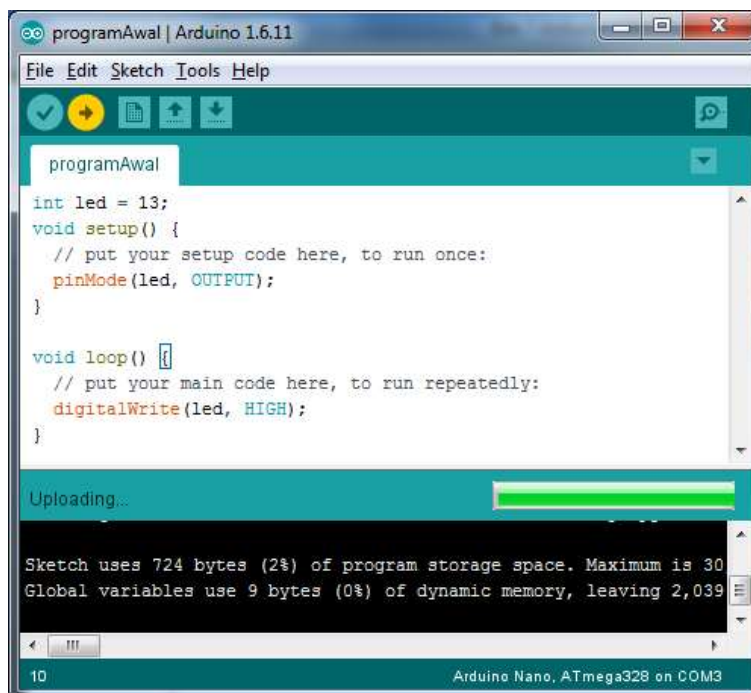
6. Setelah disimpan maka nama dari sketch akan berubah sesuai dengan nama sketch yang sudah disimpan. Lalu tuliskan kode dari program pada sketch tersebut, seperti gambar dibawah ini.



Gambar 2.6. Kode program dituliskan pada Sketch

7. Kemudian, **Compile** atau **Upload** program ke Arduino dengan memilih ikon

 untuk **Compile** dan  untuk **Upload** sketch ke Arduino.



Gambar 2.7. Proses kompilasi (compile) program Arduino

Jika terdapat error pastikan Arduino sesuai dengan jenis Board, selain itu syntax pada program juga sudah sesuai.

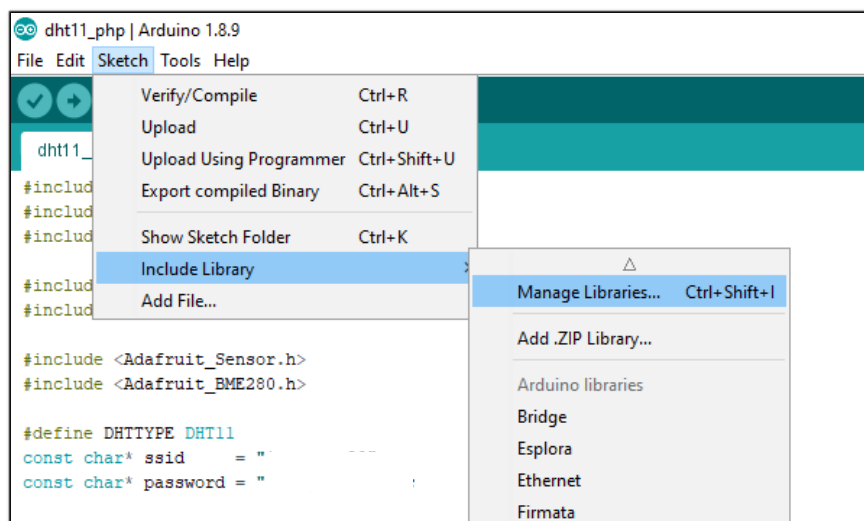
PENAMBAHAN DAN PENGGUNAAN LIBRARY DAN BOARD MANAGER TAMBAHAN

Arduino yang merupakan base board, pada penggunaannya akan banyak berinteraksi dengan modul-modul atau board lain, terutama modul atau board sensor. Beberapa sensor sudah langsung dapat dikenali dan diprogram secara default, akan tetapi sebagian sensor-sensor lainnya membutuhkan library tambahan supaya dapat diakses oleh board Arduino.

Apabila diketemukan sensor yang tidak dapat langsung digunakan maka kita bisa melakukan instalasi library tambahan, baik melalui library resmi atau library third party (pihak ketiga).

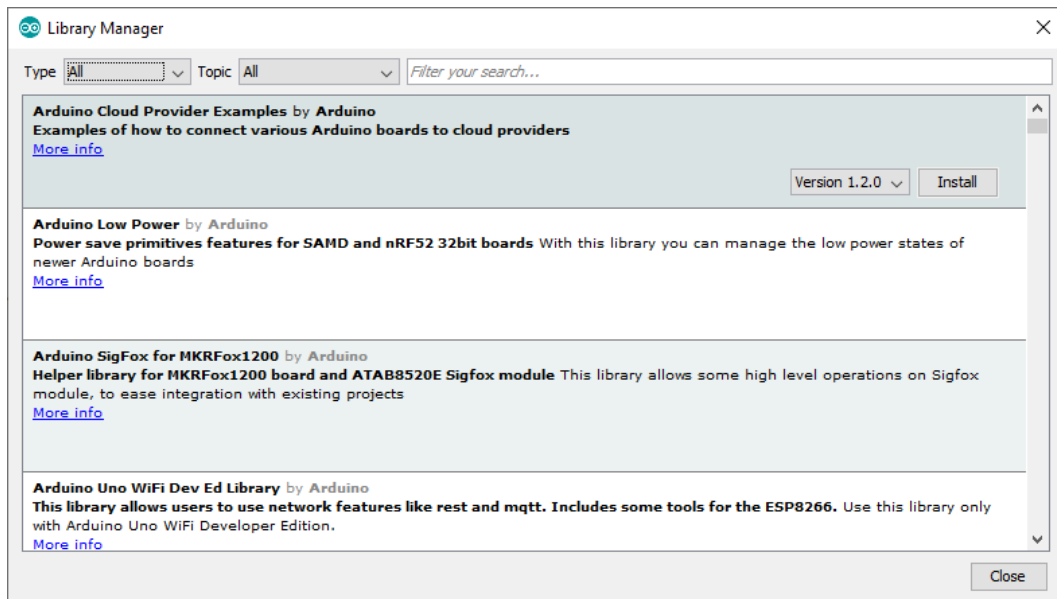
Langkah-langkah melakukan instalasi library tersebut adalah sebagai berikut.

1. Semua proses selanjutnya membutuhkan koneksi internet, pastikan komputer dihubungkan dengan internet selama melakukan proses instalasi library.
2. Buka Arduino IDE kemudian akses opsi untuk manajemen library dengan salah satu cara sebagai berikut (pilih salah satu) :
 - a. pilih menu Sketch > Include Library > Manage Libraries,
 - b. pilih menu Tools > Manage Libraries,
 - c. atau dengan menggunakan shortcut Ctrl + Shift + I



Gambar 2.8. Mengakses Manage Libraries melalui menu Sketch

3. Arduino IDE akan melakukan pemutakhiran daftar library secara online.



Gambar 2.9. Tampilan Library Manager

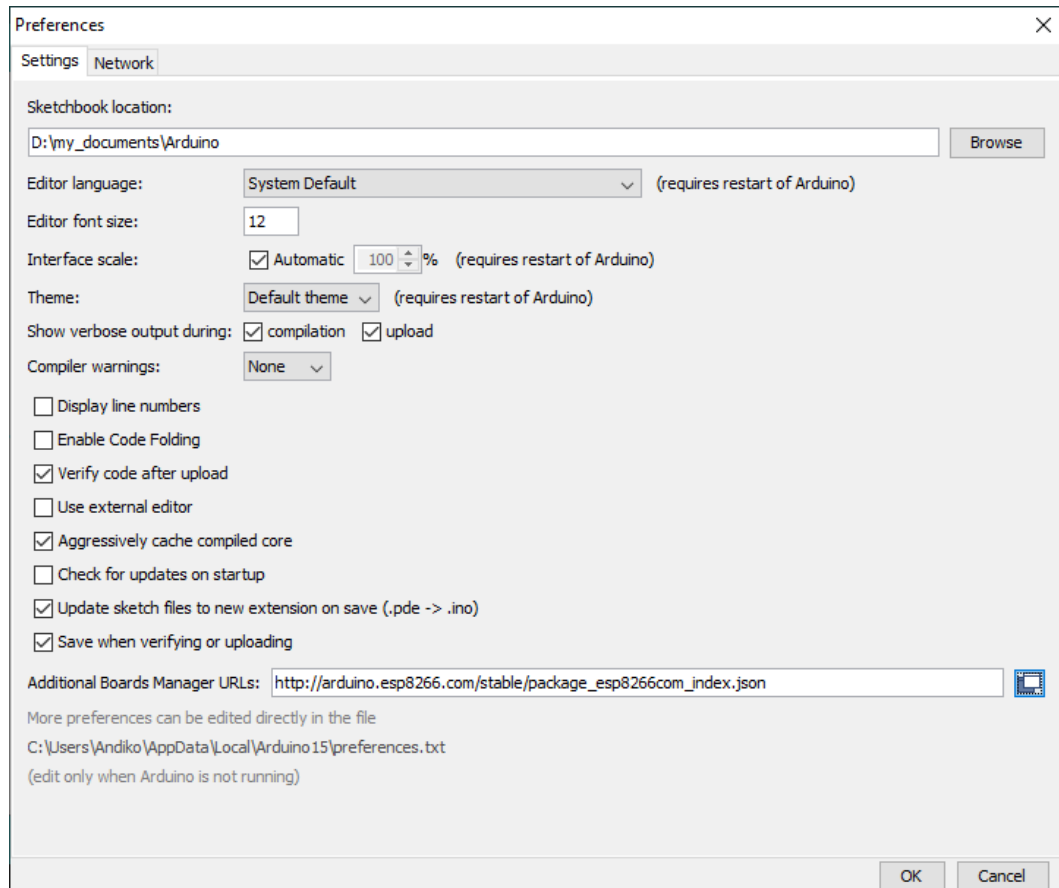
4. Setelah pemutakhiran selesai dilakukan maka akan muncul jendela Library Manger seperti terlihat pada gambar 2.2.
5. User dapat melakukan pencarian nama library yang diinginkan pada field “Filter your search” dan kemudian memilih library sesuai yang diinginkan untuk di-install. Arduino akan terlebih dahulu mengunduh berkas library yang akan di-install, pastikan komputer terhubung dengan internet seama proses instalasi berjalan.

Selain metode di atas, user juga bisa melakukan instalasi library dengan menggunakan berkas ZIP (yang didapat dari pihak ketiga) dengan cara memilih menu Sketch > Include Library > Add .ZIP Library, kemudian memilih berkas ZIP yang diinginkan.

Selain library, user juga bisa menambahkan board manager tambahan untuk menambahkan submenu Contributed Library pada menu Include Library. Seluruh library spesifik dari sebuah board yang bukan merupakan keluarga Arduino (tetapi masih merupakan keluarga AVR) akan tersedia dan dapat langsung di-install (dengan langkah install library tambahan di atas). Langkah-langkah menambahkan board manager adalah sebagai berikut.

1. Semua proses selanjutnya membutuhkan koneksi internet, pastikan komputer dihubungkan dengan internet selama melakukan proses instalasi board.
2. Pilih menu File > Preference,
3. Kemudian pada jendela yang muncul pilih tab Settings dan cari field isian dengan label “Additional Boards Manager URLs”,

4. Isikan URL dari board manager yang ingin di-install. User dapat melihat daftar board manager yang didukung oleh Arduino IDE melalui link berikut : <https://github.com/arduino/Arduino/wiki/Unofficial-list-of-3rd-party-boards-support-urls>



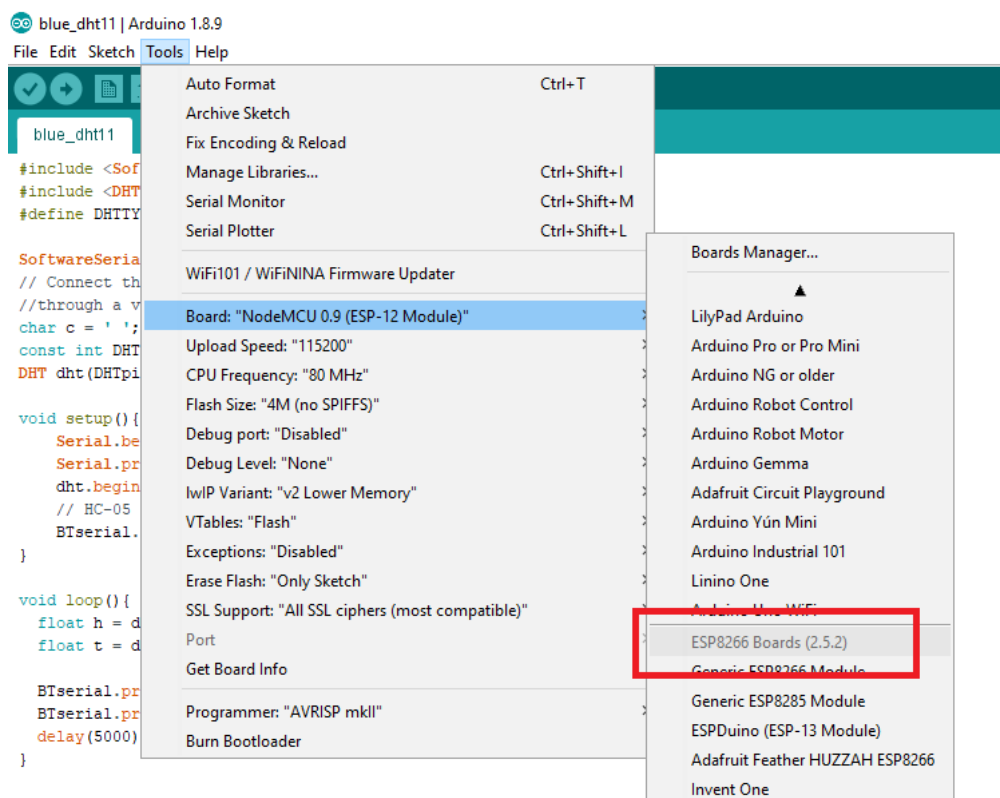
Gambar 2.10. Jendela Preferences untuk menambahkan Board Manager

5. Setelah menambahkan tautan/link di atas, langkah selanjutnya adalah melakukan instalasi dari board yang ditambahkan. Pilih menu Tools > Board > Boards Manager. Akan tampak jendela Boards Manager seperti gambar 2.11.



Gambar 2.11. Jendela Boards Manager

6. Lakukan instalasi board yang diinginkan dengan melakukan search pada field isian di bagian atas jendela. Tunggu hingga proses selesai dan lakukan pengecekan pada Tools > Board, boards baru akan muncul sebagai kelompok board terpisah seperti ditunjukkan pada gambar 2.12.



Gambar 2.12. Tampilan opsi board baru yang selesai di-install

7. Apabila akan melakukan proses compile dan upload pada board tertentu (termasuk ESP8266) untuk pertama kalinya kita harus melakukan penyesuaian setting board (Tools > Board). Untuk ESP8266 (NodeMCU) ada 2 macam jenis board yang tersedia yaitu “NodeMCU 0.9 (ESP-12 module)” atau “NodeMCU 1.0 (ESP-12E module)”. Jika menggunakan ESP-01 maka jenis board yang dipilih adalah “Generic ESP8266 module”

MODUL 3

BASIC INPUT OUTPUT

APLIKASI BASIC I/O PADA LAMPU LED

LED atau singkatan dari Light Emitting Diode adalah salah satu komponen elektronik yang tidak asing lagi di kehidupan manusia saat ini. LED saat ini sudah banyak dipakai, seperti untuk penggunaan lampu permainan anak-anak, untuk rambu-rambu lalu lintas, lampu indikator peralatan elektronik hingga ke industri, untuk lampu emergency, untuk televisi, komputer, pengeras suara (speaker), hard disk eksternal, proyektor, LCD, dan berbagai perangkat elektronik lainnya sebagai indikator bahwa sistem sedang berada dalam proses kerja, dan biasanya berwarna merah atau kuning. LED ini banyak digunakan karena konsumsi daya yang dibutuhkan tidak terlalu besar dan beragam warna yang ada dapat memperjelas bentuk atau huruf yang akan ditampilkan. dan banyak lagi.

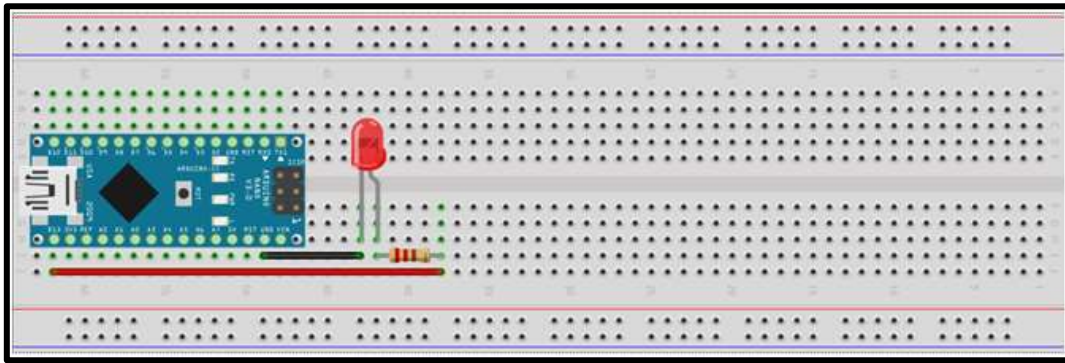
Pada dasarnya LED itu merupakan komponen elektronika yang terbuat dari bahan semi konduktor jenis dioda yang mampu memancarkan cahaya. LED merupakan produk temuan lain setelah dioda. Strukturnya juga sama dengan dioda, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan P-N. Untuk mendapatkan emisi cahaya pada semikonduktor, doping yang pakai adalah galium, arsenic dan phosporus. Jenis doping yang berbeda menghasilkan warna cahaya yang berbeda pula.

1. ARDUINO LED

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. LED
4. Resistor
5. Kabel Jumper

b. RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 3.1 Rangkaian LED

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(3, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(100);
}
```

Modul 3.1 Program LED

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasangkan kabel jumper ke pin 13, lalu ujung lainnya sambungkan ke resistor.
4. Lalu pasang LED bagian yang panjang ke resistor.
5. Pasang kabel jumper ke ground pasangkan ke bagian pendek LED.
6. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
7. Amati apa yang terjadi.

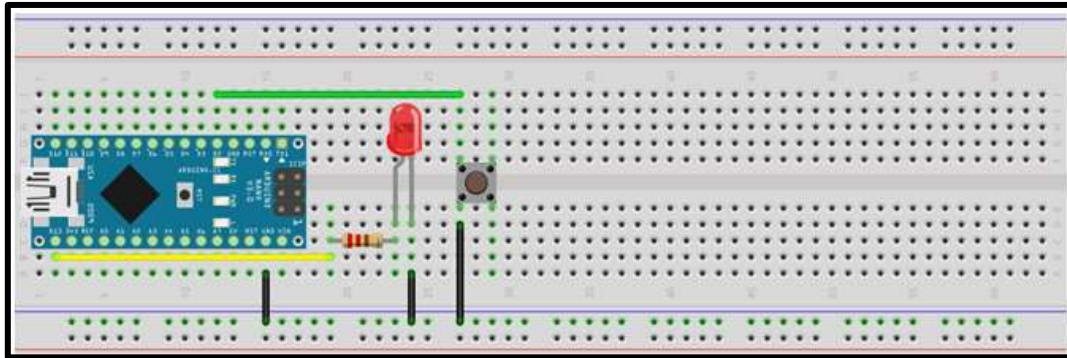
2. ARDUINO LED DAN BUTTON

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. LED
4. Resistor

5. Kabel Jumper
6. Button

b. RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 3.2 Rangkaian LED dan Button

```
int x, led = 13, btn = 2;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(btn, INPUT_PULLUP);
}

void loop() {
  x = digitalRead(btn);
  if(x==0){
    digitalWrite(led, HIGH);
    delay(100);
  }else{
    digitalWrite(led, LOW);
    delay(100);
  }
}
```

Modul 3.2 Program LED dan Button

Langkah – langkahnya :

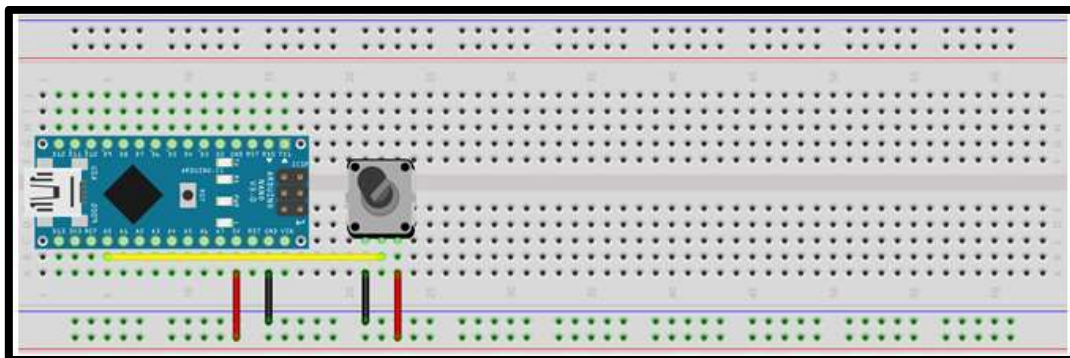
1. Tulis program diatas di sketch Arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasangkan kabel jumper ke pin 13, lalu ujung lainnya sambungkan ke resistor.
4. Lalu pasang LED bagian yang panjang ke resistor.
5. Pasang kabel jumper ke ground pasangkan ke bagian pendek LED.
6. Pasang button ke project board lalu pasangkan ke pin 2 dan ground menggunakan kabel jumper.
7. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
8. Amati apa yang terjadi.

3. ARDUINO ANALOG DIGITAL CONVERTER

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. Potensio
4. Kabel Jumper

b. RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 3.3 Rangkaian ADC dengan Potensio

```
int adc;

void setup() {
  Serial.begin(9600);
}

void loop() {
  adc = analogRead(A0);
  Serial.println(adc);
  delay(100);
}
```

Modul 3.3 Program ADC dengan Potensio

Langkah – langkahnya :

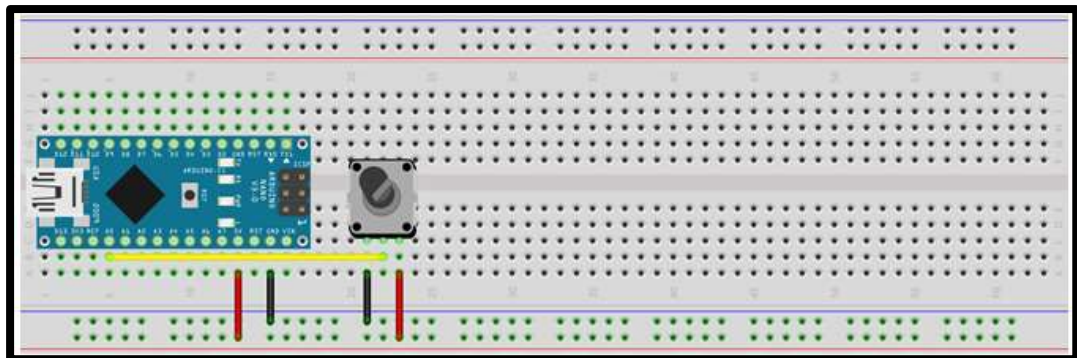
1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasang Potensio ke project board.
4. Pasang kabel jumper ke pin A0, 5v, dan ground yang ada di arduino.
5. Pasangkan kabel jumper ke potensio.
6. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
7. Buka serial monitor arduino dan putar potensio nya.
8. Amati apa yang terjadi.

4. ARDUINO SEVEN SEGMENT

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. 7 Segment
4. Kabel Jumper

b. RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 3.4. Rangkaian Seven Segmen

```
void setup() {  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(8, OUTPUT);  
  pinMode(9, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(3,1);  
  digitalWrite(4,1);  
  digitalWrite(5,1);  
  digitalWrite(6,1);  
  digitalWrite(7,1);  
  digitalWrite(8,1);  
  digitalWrite(9,1);  
}
```

Modul 3.4. Program Seven Segmen

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasang 7 Segment ke project board.
4. Pasang kabel jumper ke pin yang ada tertera di gambar rangkaian ke arduino.
5. Pasangkan kabel jumper ke 7 segmen.
6. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.

7. Amati apa yang terjadi.

TUGAS PRAKTIKUM

1. Buatlah program untuk LED berjalan menggunakan 4 LED lalu buat rangkaiannya dan coba implementasikan !

MODUL 4

KOMUNIKASI SERIAL

Komunikasi data merupakan bentuk komunikasi yang secara khusus berkaitan dengan transmisi atau pemindahan data antara komputer-komputer, komputer dengan piranti-piranti yang lain dalam bentuk data digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu masyarakat informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain.

Komunikasi Data saat ini menjadi bagian dari kehidupan masyarakat, karena telah diterapkan dalam berbagai bentuk aplikasi misal: komunikasi antar komputer yang populer dengan istilah internet, Handphone ke komputer, Handphone ke Handphone, komputer atau handphone ke perangkat lain misal: printer, fax, telpon, camera video, dan lain-lain.

Model komunikasi sederhana terdiri dari :

- Source
- Transmitter (Pengirim)
- Transmission System (Sistem Transmisi)
- Receiver (Penerima)
- Destination (Tujuan)

Source

Alat ini membangkitkan data sehingga dapat ditransmisikan, contoh alat yang bisa disebut source adalah : telepon dan PC (Personal Computer)

Transmitter (Pengirim)

Biasanya data yang dibangkitkan dari sistem sumber tidak ditransmisikan secara langsung dalam bentuknya aslinya. Sebuah trasmitter cukup memindah dan menandai informasi dengan cara yang sama seperti menghasilkan sinyal-sinyal elektromagnetik yang dapat ditransmisikan melewati beberapa sistem transmisi berurutan.

Contoh : Sebuah modem tugasnya menyalurkan suatu digital bit stream dari suatu alat yang sebelumnya sudah dipersiapkan misalnya PC (Personal Computer), dan mentransformasikan bit stream tersebut menjadi suatu sinyal analog yang dapat melintasi jaringan telepon.

Transmission System (Sistem Transmisi)

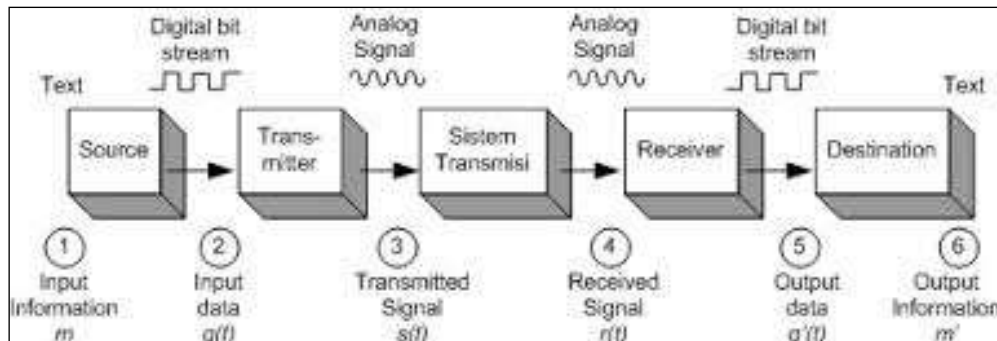
Sistem transmisi berupa jalur transmisi tunggal (single transmission line) atau jaringan kompleks (complex network) yang menghubungkan antara sumber dengan tujuan (destination).

Receiver (Penerima)

Receiver menerima sinyal dari sistem transmisi dan menggabungkannya ke dalam bentuk tertentu yang dapat ditangkap oleh tujuan. Contoh : modem akan menerima sinyal analog yang datang dari jaringan atau jalur transmisi dan mengubahnya menjadi suatu digital bit stream.

Destination (Tujuan)

Destination menangkap data yang dihasilkan oleh receiver



Gambar 4.1 Alur data ke tujuan

Pada sistem komunikasi terdapat dua jenis komunikasi yaitu komunikasi paralel dan komunikasi serial.

1. Komunikasi paralel merupakan komunikasi yang menggunakan konsep pengiriman data secara paralel (bersamaan), semua bit data pada satu waktu sekaligus. Jalur yang dibutuhkan cukup banyak namun memiliki kecepatan yang cukup cepat dibandingkan komunikasi serial.
2. Komunikasi serial menggunakan konsep komunikasi pengiriman data secara serial (berurutan), satu demi satu bit secara bergantian. Komunikasi

jenis ini walaupun lebih lambat dari komunikasi paralel namun jalur yang digunakan lebih sedikit.

Komunikasi Serial Arduino

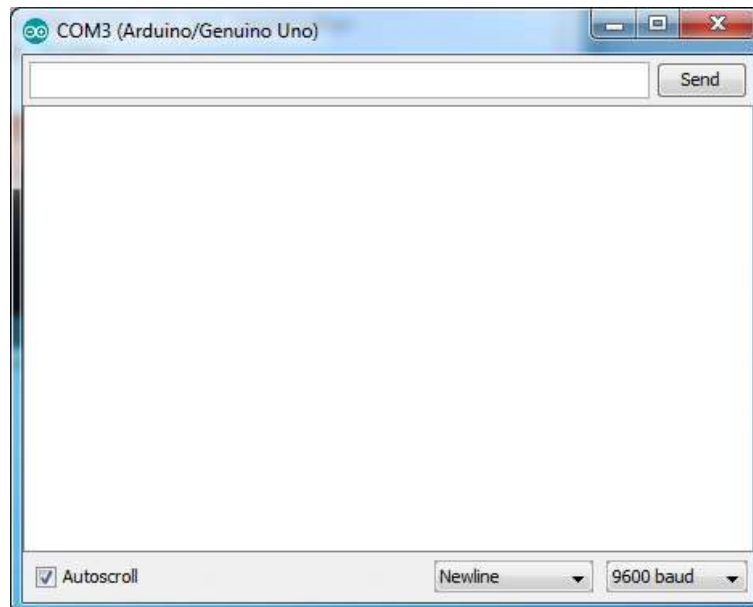
Pada Arduino terdapat PIN yang berfungsi untuk komunikasi data serial, untuk pengiriman data yaitu menggunakan Transmitter (Tx), sedangkan untuk menerima data yaitu menggunakan Receiver (Rx). Dalam komunikasi data serial Arduino dipermudah dengan adanya fungsi-fungsi yang sudah disediakan dari Arduino, diantaranya yaitu:

Tabel 4.1. Fungsi Komunikasi Serial Arduino

if(Serial)	:	Untuk mengecek kesiapan Port
Serial.available()	:	Untuk mengecek data pada Buffer Penerima
Serial.begin()	:	Untuk mengeset kecepatan Transmisi Data
Serial.end()	:	Untuk menon-aktifkan PIN Rx dan Tx sebagai fungsi serial dan kembali sebagai PIN Input atau Output
Serial.find()	:	Untuk mencari string dalam Buffer Data
Serial.findUntil()	:	Untuk mencari Buffer data sampai panjang data sesuai dengan sama yang ditemukan
Serial.flush()	:	Untuk menunggu data terkirim semua
Serial.parseFloat()	:	Untuk mengambil data float pertama dari data di Buffer Serial
Serial.parseInt()	:	Untuk mengambil data integer pertama dari data di Buffer Serial
Serial.peek()	:	Untuk mengambil data berikutnya di Buffer Penerima
Serial.print()	:	Untuk mengirim data ASCII
Serial.println()	:	Untuk mengirim data ASCII + CR,LF (kode Enter)
Serial.read()	:	Untuk membaca data yang diterima
Serial.readBytes()	:	Untuk membaca data Byte yang diterima
Serial.readBytesUntil()	:	Untuk membaca data Byte yang diterima sampai ditemukan data yang sesuai
Serial.setTimeout()	:	Untuk menatur batas maksimum waktu tunggu (timeout) transmisi data
Serial.write()	:	Untuk mengirim data byte (numerik)
Serial.serialEvent()	:	Untuk interupsi pengiriman data serial

Selain fungsi-fungsi untuk komunikasi data serial, pada Arduino juga menyediakan Serial Monitor untuk melihat pengiriman data yang ada pada Arduino disertai jenis transmisi yang digunakan atau kecepatan pengiriman data serial. Untuk menggunakan Serial Monitor tersebut dapat dibuka dengan cara Pilih **Menu Bar Tools >> Pilih Serial Monitor** atau dengan cara cepat pada keyboard dengan menekan tombol **Ctrl + Shift + M**. Sebelum menggunakan Serial Monitor, yang perlu

diperhatikan terlebih dahulu yaitu kondisi Arduino harus hidup yang mana terhubung ke PC atau laptop dengan Aplikasi Arduino. Setelah dibuka dengan cara tersebut maka akan muncul tampilan jendela dari Serial Monitor seperti berikut.



Gambar 4.1. Tampilan Serial Monitor Arduino

Untuk komunikasi data serial dasar dapat dicoba dengan membuat program dibawah ini.

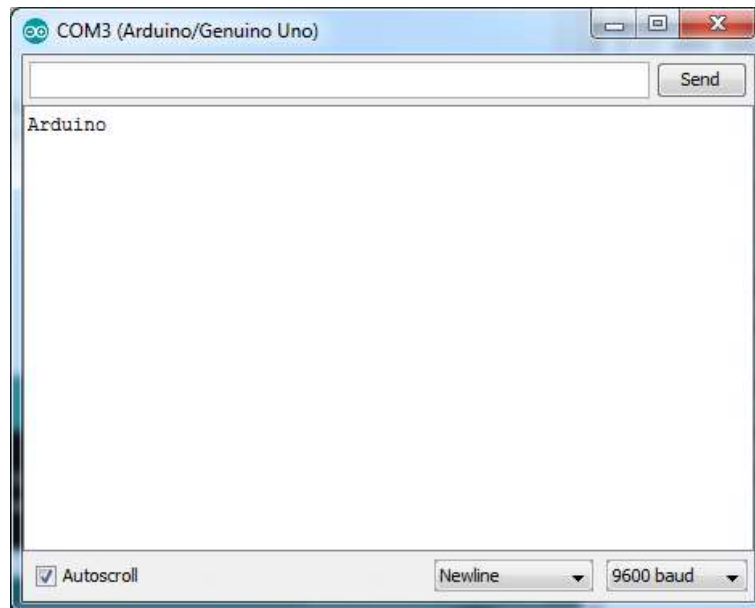
```
String nama = "Arduino";

void setup()
{
  // set baudrate 9600
  Serial.begin(9600);
}

void loop()
{
  // data yg diterima dikirim kembali.
  Serial.print(nama);
  delay(500);
}
```

Modul 4.1. Pengiriman Data Serial Sederhana

Dari program diatas pengiriman data menggunakan baudrate 9600. Program diatas yaitu untuk mengirim data berupa string dengan nama variabel nama yang memiliki nilai string yaitu Arduino. Data akan dikirim setiap 500 milidetik, untuk hasil dari program bisa dilihat pada gambar berikut.



Gambar 4.2. Tampilan Pengiriman data dilihat dari Serial Monitor Arduino

Pada komunikasi data serial perlu dideklarasikan Baudrate dari pengiriman data tersebut, terdapat banyak jenis baudrate untuk pengiriman data yang tersedia antara lain 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200. Untuk jenis Arduino memiliki batasan baudrate yang berbeda, sehingga ketika ingin menggunakan komunikasi data serial diharapkan tau spesifikasi dari Arduino yang digunakan. Semakin besar baudrate yang digunakan maka akan semakin cepat pengiriman datanya.

```
int angkaint = 123;
float angkafloat 123.5;
char datachar = 123;

void setup()
{
    // set baudrate = 19200
    Serial.begin(19200);
}

void loop()
{
    // kirim string/text
    Serial.print(""apa kabar"");

    // kirim tipe angka integer 123
    Serial.println(angkaint);

    // kirim angka type float 123.5
    Serial.println(angkafloat);

    // kirim code ascii 123 yaitu tanda kurung buka {
    Serial.println(datachar);
}
```



```

// kirim angka dlm bentuk heksa , akan dikirim 7BH
Serial.println(angkaInt,HEX);

// kirim angka dlm bentuk biner , 0111 1011
Serial.println(angkaInt,BIN);
}

```

Modul 4.2. Contoh Pengiriman Data Serial dengan Tipe Data

Pada saat pengiriman data terdapat interupsi yang dapat mengeksekusi data secara langsung ketika terdapat data masuk. Ketika terdapat data maka yang akan dieksekusi yaitu fungsi `serialEvent()`. Berikut contoh interupsi data pada pengiriman data serial.

```

String inputString = "";
boolean stringComplete = false;

void setup()
{
  Serial.begin(9600);
  inputString.reserve(200);
}

void loop()
{
  if (stringComplete)
  {
    Serial.println(inputString);
    inputString = "";
    stringComplete = false;
  }
}

void serialEvent()
{
  while (Serial.available())
  {
    char inChar = (char)Serial.read();
    inputString += inChar;
    if (inChar == '\n')
    {
      stringComplete = true;
    }
  }
}

```

Modul 4.3. Contoh Pengiriman Data Serial dengan Interupsi

TUGAS PRAKTIKUM

1. Buatlah program yang bisa mencetak karakter set ASCII dan bentuk desimalnya dimulai dari **whitespace (spasi) sampai tilde (~)**. Susunlah output serial program seperti contoh berikut.

```
...  
A = 65  
B = 66  
...
```

Pastikan program hanya menampilkan whitespace hingga tilde saja (artinya apabila sampai tilde maka kembali lagi menampilkan whitespace dan berulang seterusnya)

MODUL 5

TIMER DAN INTERRUPT

Timer & Counter merupakan fitur yang telah tertanam di mikrokontroler AVR yang memiliki fungsi terhadap waktu. Fungsi pewaktu yang dimaksud disini adalah penentuan kapan program tersebut dijalankan, tidak hanya itu saja fungsi timer yang lainnya adalah PWM, ADC, dan Oscillator. Prinsip kerja timer dengan cara membagi frekuensi (prescaler) pada clock yang terdapat pada mikrokontroler sehingga timer dapat berjalan sesuai dengan frekuensi yang di kehendaki.

Timer merupakan fungsi waktu yang sumber clocknya berasal dari clock internal. Sedangkan counter merupakan fungsi perhitungan yang sumber clocknya berasal dari external mikrokontroler. Salah satu contoh penggunaan fungsi timer yaitu pada jam digital yang sumber clocknya bisa menggunakan crystal oscillator dan contoh penggunaan counter pada penghitung barang pada konveyor yang sumber clocknya berasal dari sensor yang mendeteksi barang tersebut.

Pada Arduino AVR yang digunakan yaitu Atmel AVR Atmega 168 atau Atmega328. Kedua chip tersebut kompatibel tetapi memiliki memori internal yang berbeda, walaupun keduanya memiliki 3 timer yang disebut dengan Timer0, Timer1, dan Timer2. Timer0 dan Timer2 memiliki waktu 8 bit, sedangkan untuk Timer1 memiliki waktu 16 bit. Antara 8 bit dengan 16 bit hanya berbeda pada resolusi waktu, karena 8 bit hanya memiliki 256 nilai, sedangkan 16 bit memiliki 65536 nilai yang berarti resolusi waktu lebih tinggi.

Timer0

Timer0 yaitu timer 8 bit. Pada Arduino timer ini digunakan untuk fungsi timer seperti delay(), millis(), dan micros(). Selain itu terdapat timer0 register dimana dapat diubah, akan tetapi timer0 register mempengaruhi fungsi timer Arduino.

Timer1

Timer1 yaitu timer 16 bit. Pada Arduino Uno timer1 biasa digunakan untuk library Servo, sedangkan untuk Arduino Mega pada timer5.

Timer2

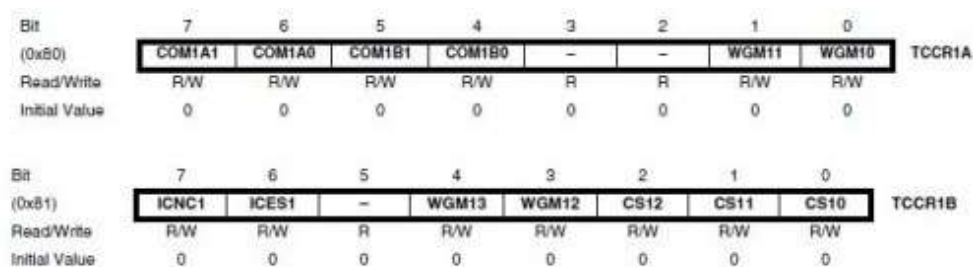
Timer2 yaitu timer 8 bit, seperti timer0. Timer2 biasa digunakan untuk fungsi nada atau tone().

Timer3, Timer4, Timer5

Timer 3, 4, 5 hanya tersedia pada Arduino mega dimana semua Timer memiliki waktu 16 bit.

Timer Register

Timer register mempengaruhi Timer yang lainnya, timer ini dapat diubah atau dikonfigurasi.



- TCCR_x** - Timer/Counter register. Berfungsi untuk mengatur Prescale.
- TCNT_x** - Timer/Counter register. Untuk menyimpan nilai timer.
- OCR_x** - Keluaran pembanding register
- ICR_x** - Masukan untuk register (hanya terdapat pada timer 16 bit)
- TIMSK_x** - Timer/Counter untuk mengaktifkan/menon-aktifkan interupsi timer
- TIFR_x** - Timer/Counter untuk menunjukkan timer interupsi yang tertunda

Interupsi

Interupsi merupakan kondisi dimana terdapat masukan pada saat program berjalan. Sebuah interupsi dapat langsung dijalankan ketika terdapat kondisi interupsi khusus. Jika interupsi dijalankan maka program yang sedang berjalan akan menunggu sampai interupsi tersebut selesai dikerjakan.

Interupsi pada umumnya dapat diaktifkan/ dinonaktifkan dengan cara interrupts()/noInterrupts(). Secara default pada Arduino interupsi diaktifkan, penanda interupsi diaktifkan/dinonaktifkan dengan pengaturan bit pada TIMSK_x (The Interrupt Mask Register).

Ketika terjadi interupsi, maka interupsi akan ditetapkan pada TIFR_x (The Interrupt Flag Register) sebagai penanda dari interupsi. Jika menggunakan ISR (Interrupt Service Routines) setelah tidak dipakai maka akan terhapus secara otomatis atau dapat juga dihapus secara manual pada TIFR_x.

Pada Arduino Uno dan Arduino Nano memiliki 2 PIN interrupt yaitu INT0 pada PIN 2 digital dan INT1 pada PIN 3 digital.

Untuk membuat interupsi pada Arduino dapat menggunakan fungsi berikut.

```
attachInterrupt(interrupt, function, mode);
```

Pada bagian function dapat diisi dengan program yang ingin dijalankan saat PIN interrupt dipicu. Kemudian, terdapat 4 mode yang dapat memicu interupsi, diantaranya :

- a. Mode LOW, pada mode ini interrupt akan diaktifkan saat PIN INT memiliki logika LOW dan akan aktif selama masih berlogika LOW.
- b. Mode CHANGE, pada mode ini interrupt akan diaktifkan saat terjadi perubahan logika baik dari LOW ke HIGH ataupun HIGH ke LOW. Jalannya program hanya bersifat sementara, ketika program pada interrupt selesai maka akan melanjutkan program sebelumnya.
- c. Mode RISING, pada mode ini sama seperti mode CHANGE yaitu interrupt akan aktif ketika terjadi perubahan logika dari LOW ke HIGH, dan akan kembali ke program sebelumnya setelah selesai proses interrupt.
- d. Mode FALLING, pada mode ini juga sama seperti mode CHANGE yaitu interrupt akan aktif ketika terjadi perubahan logika dari HIGH ke LOW, dan juga akan kembali ke program sebelumnya setelah selesai proses interrupt.

PWM dan Timer

Hubungan antara timer dengan PWM sangatlah terlihat, jika kita lihat dari datasheet terdapat PIN yang memiliki data OCRxA, OCRxB, atau OCRxC dimana x berarti interval waktu antara 0 sampai dengan 5. Fungsi PWM saling berkaitan dengan fungsi PIN yang lainnya.

Arduino UNO memiliki 3 Timer dan 6 PIN keluaran PWM. Diantaranya yaitu,

- PIN 5 dan 6, dikendalikan oleh Timer0
- PIN 9 dan 10, dikendalikan oleh Timer1
- PIN 11 dan 3, dikendalikan oleh Timer2

Penggunaan Timer Overflow Untuk LED Blinking

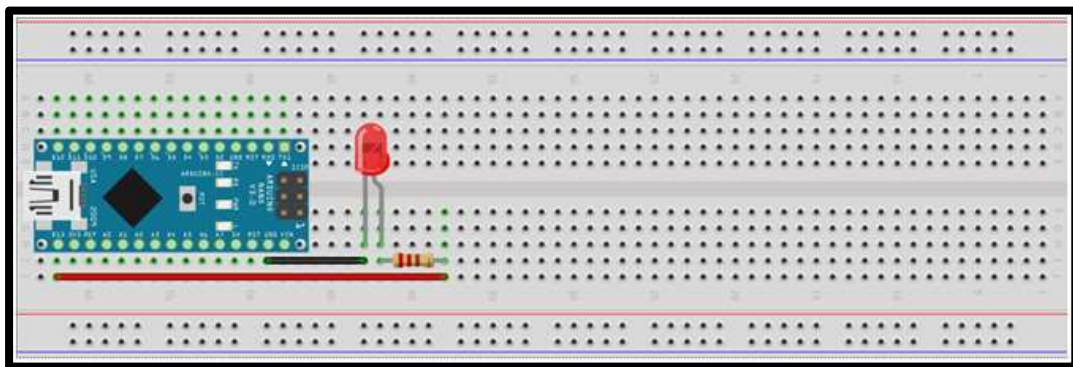
Penggunaan Timer untuk membuat led berkedip setiap 500 milidetik. Pada dasarnya LED akan berubah kondisi setiap terjadi interupsi timer overflow. Nilai

register timer di-set sedemikian rupa sehingga menghasilkan waktu overflow setiap 500 milidetik.

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. LED
4. Resistor
5. Kabel Jumper

b. RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 5.1. Rangkaian LED Timer Overflow

```
boolean dataLed = 0;
unsigned int overflowCount = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 34286;
  TCCR1B |= (1 << CS12); //256 prescaler
  TIMSK1 |= (1 << TOIE1);
  sei();
}

ISR(TIMER1_OVR_vect)
{
  TCNT1 = 34286;
  if(dataLed == 0){
    dataLed = 1;
  }else{
    dataLed = 0;
  }
  digitalWrite(13, dataLed);
  overflowCount++;
}
```

```

}

void loop()
{
  Serial.print("Overflow count = ");
  Serial.println(overflowCount, DEC);
  delay(100);
}

```

Modul 5.1. Program Timer Overflow

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasangkan kabel jumper ke pin 13, lalu ujung lainnya sambungkan ke resistor.
4. Lalu pasang LED bagian yang panjang ke resistor.
5. Pasang kabel jumper ke ground pasangkan ke bagian pendek LED.
6. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
7. Amati apa yang terjadi.

Counter Mode Falling Edge

Program ini dibuat untuk mengetahui banyaknya penekanan switch atau button pada input counter. Konfigurasi counter menggunakan mode falling edge.

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board

b. LANGKAH PRAKTIKUM

```

void setup()
{
  Serial.begin(9600);
  TCCR1A=0x00;
  TCCR1B=0x07;
  TCNT1H=0x00;
  TCNT1L=0x00;
  ICR1H=0x00;
  ICR1L=0x00;
  OCR1AH=0x00;
  OCR1AL=0x00;
  OCR1BH=0x00;
  OCR1BL=0x00;
}

void loop()

```

```

{
  Serial.print("Nilai TCNT1 = ");
  Serial.println(TCNT1, DEC);
  delay(100);
}

```

Modul 5.2. Program Counter Falling Edge

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
4. Amati apa yang terjadi pada Serial Monitor.

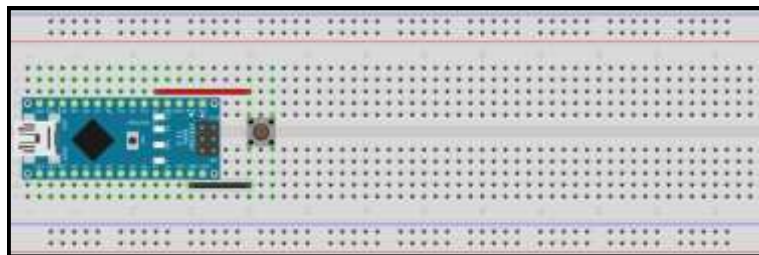
Interupsi Input

Program berikut untuk memberikan contoh penggunaan interupsi untuk input.

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. Push Button
4. Kabel Jumper

b. LANGKAH PRAKTIKUM



Gambar 5.2. Rangkaian LED Timer Overflow

```

unsigned long counts;
unsigned long previousMillis;

void impulse() { // dipanggil setiap ada sinyal FALLING
di pin 2
  counts++;
}

#define LOG_PERIOD 1000 // cetak tiap detik

void setup() { //setup subprocedure
  counts = 0;
  Serial.begin(9600);
  pinMode(2, INPUT);
}

```



```

    digitalWrite(2, HIGH); // mengaktifkan internal pull up
    //define external interrupts
    attachInterrupt(digitalPinToInterrupt(2), impulse,
    FALLING);
    Serial.println("Start counter");
}

void loop() { //main cycle
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis > LOG_PERIOD) {
        previousMillis = currentMillis;
        Serial.println(counts);
        counts = 0;
    }
}

```

Modul 5.3. Program Interupsi dengan Push Button

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasang Push button pada breadboard.
4. Hubungkan kabel jumper dari PIN 2 digital ke push button.
5. Hubungkan kabel jumper dari push button ke ground (GND).
6. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
7. Amati apa yang terjadi pada Serial Monitor.

BAB 6

PENGEMBANGAN LANJUT ARDUINO

SENSOR DHT11

DHT11 merupakan modul sensor elektronik yang dapat mendeteksi suhu dan kelembaban suatu ruangan secara akurat dan mengeluarkan data yang terdeteksi berupa signal digital. Kelebihan sensor ini dapat diandalkan dan memiliki kestabilan yang lama dalam pengoperasiannya, respon yang cepat dalam pendeteksi suhu dan kelembaban, pengukuran suhu dan kelembaban yang tepat. DHT11 dapat digunakan pada mikrokontroler yang memproses 8 bit data.

DHT11 menggunakan Single Bus Communication yang berarti hanya menggunakan satu jalur untuk komunikasi ke sensor tersebut. Sensor ini akan mengirim kan 40 bit data secara berurutan dengan format 8 bit data integer kelembaban + 8 bit data decimal kelembaban + 8 bit data integer suhu + 8 bit data fractional suhu + 8 bit data checksum.

Contoh :

40 bit data diterima

0011 0101 0000 0000 0001 1000 0000 0000 0100 1101

High Humidity Low Humidity High Temperature Low Temperature Parity Check

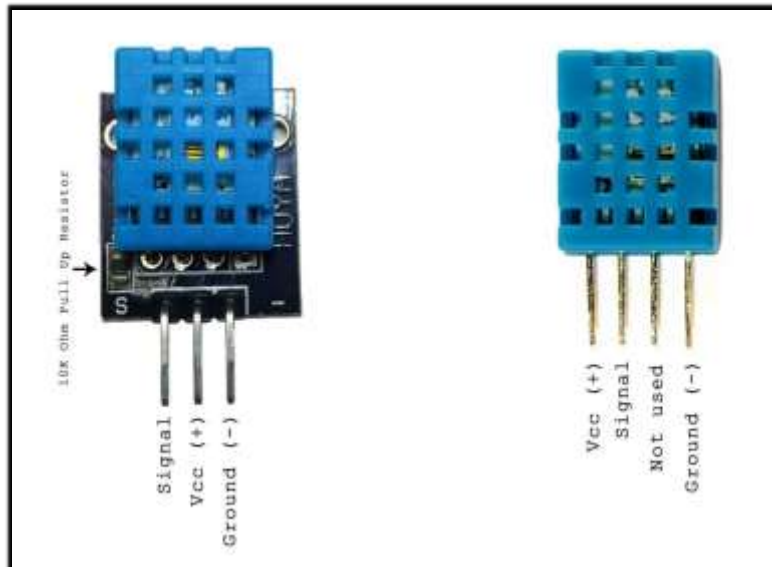
Checksum :

0011 0101 + 0000 0000 + 0001 1000 + 0000 0000 = 0100 1101

Data yang diterima benar :

Kelembaban : 00 11 0101 = 53% RH

Suhu : 00011000 = 24° C



Gambar 6.1. Sensor DHT11 (3 pin dan 4 pin)

Pada umumnya bisa ditemui 2 tipe sensor DHT11 yaitu yang menggunakan 3 pin dan 4 pin. Modul DHT11 yang memiliki 4 pin membutuhkan tambahan resistor pull-up untuk memastikan level sinyal yang dikirim dan diterima selalu pada level High dalam kondisi default. Besar hambatan resistor pull-up yang biasa digunakan adalah 10K ohm dan dipasangkan menghubungkan pin VCC (+5V) dan Signal. Sedangkan untuk DHT11 yang memiliki 3 pin, resistor pull up sudah terintegrasi pada modul sehingga tidak perlu lagi ditambahkan resistor di antara kaki VCC dan Signal.

a. ALAT DAN BAHAN

1. Arduino Nano
2. Sensor DHT11
3. Project Board
4. Kabel Jumper

b. LANGKAH PRAKTIKUM

Sebelum membuat program Arduino Nano dan DHT11 langkah awal yang perlu diperhatikan yaitu mencari file library untuk DHT11, karena pada Arduino tidak disertakan dan harus download. Ketika sudah didownload dan diimport pada sketch Arduino, maka langsung saja ke langkah perangkaian. Untuk download library bisa

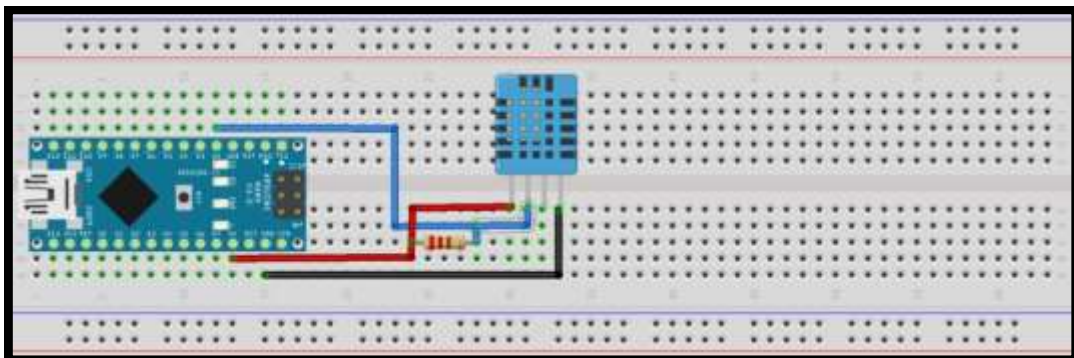
langsung download pada alamat berikut: <https://github.com/adafruit/DHT-sensor-library>.

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup(){
  Serial.begin(9600);
  Serial.println("DHT11 Output!");
  dht.begin();
}

void loop(){
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if(isnan(t) || isnan(h)){
    Serial.println("Failed to read DHT11");
  }else{
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print(" %\t");
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.println(" *C");
  }
}
```

Modul 6.1. Program Arduino dengan Sensor DHT11



Gambar 6.2. Rangkaian Arduino Nano dan DHT11

Langkah – langkahnya :

1. Tulis program diatas di sketch Arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasangkan kabel jumper ke pin 2, lalu ujung lainnya sambungkan ke resistor, dan paralelkan pin ke bagian pi data pada Sensor DHT11 .
4. Lalu pasang kabel jumper dari daya 5V ke resistor, dan ujung lainnya sambungkan ke pin + pada Sensor DHT11.

5. Pasang kabel jumper ke ground pasang ke bagian pin – pada Sensor DHT11.
6. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
7. Amati apa yang terjadi pada Serial Monitor Arduino.

SENSOR ULTRASONIK

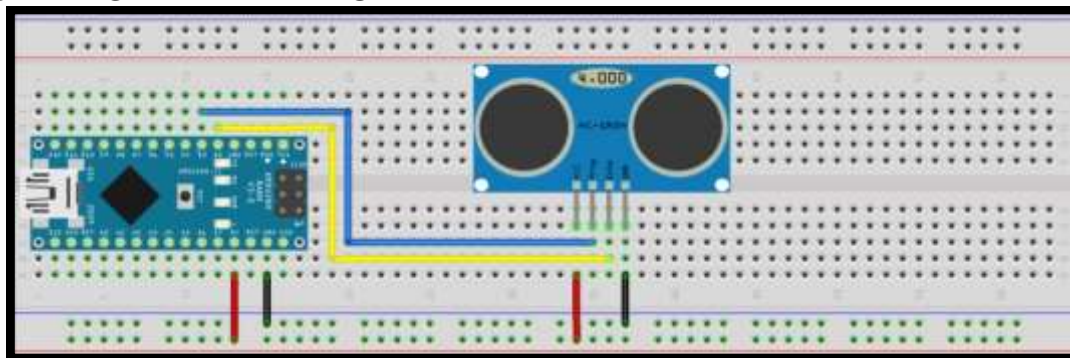
Sensor Ultrasonik merupakan komponen yang prinsip kerjanya memantulkan gelombang suara sehingga dapat dideteksi jarak benda berdasarkan pantulan tersebut. Frekuensi sensor ultrasonik berkisar antara 40 KHz sampai 400 KHz.

Sensor ultrasonik terdiri dari dua bagian, yaitu bagian pengirim data dan bagian penerima data. Data yang dikirim akan dipantulkan pada objek didepannya, sehingga ketika terdapat sebuah objek gelombang ultrasonik akan memantul dan diterima oleh penerima data. Cara menghitung jarak dari sensor ultrasonik yaitu mengalikan separuh waktu yang dipakai oleh signal ultrasonik untuk berjalan dari rangkaian Tx (Transmitter) hingga ditangkap kembali oleh rangkaian Rx (Receiver), dengan kecepatan signal ultrasonik tersebut pada media rambat yang dipakai yaitu udara.

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. HC-SR04
4. Kabel Jumper

b. LANGKAH PRAKTIKUM



Gambar 6.3. Rangkaian Arduino Nano dengan Sensor HC-SR04

```

const int pTrig = 3;
const int pEcho = 2;
long durasi = 0;
long jarak = 0;

void setup() {
  Serial.begin(9600);
  pinMode(pTrig, OUTPUT);
  pinMode(pEcho, INPUT);
}

void loop() {
  //trigger selama 10us
  digitalWrite(pTrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(pTrig, LOW);

  durasi = pulseIn(pEcho, HIGH);
  jarak = (durasi * 0.034)/2;

  Serial.print("Jarak : ");
  Serial.print(jarak);
  Serial.println(" Cm.");

  delay(10);
}

```

Modul 6.2. Program Arduino dengan Sensor DHT11

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasangkan kabel jumper arduino sesuai dengan pin yang ada di rangkaian.
4. Lalu pasang kabel jumper yang sudah terpasang di pin arduino ke sensor dengan hati hati.
5. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
6. Buka serial monitor, gerakkan benda didepan sensor dengan menjauh dan mendekat.
7. Amati apa yang terjadi.

SENSOR PIR

Sensor PIR (Passive Infra Red) merupakan sensor yang digunakan untuk mendeteksi pancaran sinar infra merah. Sensor PIR bersifat pasif, karena sensor ini menerima pancaran sinar infra merah dari luar dan sensor ini tidak memancarkan sinar infra merah. Salah satu contohnya yaitu manusia, dimana memiliki suhu tubuh

yang bersifat panas, maka akan memberikan infra merah dan ditangkap oleh sensor PIR.

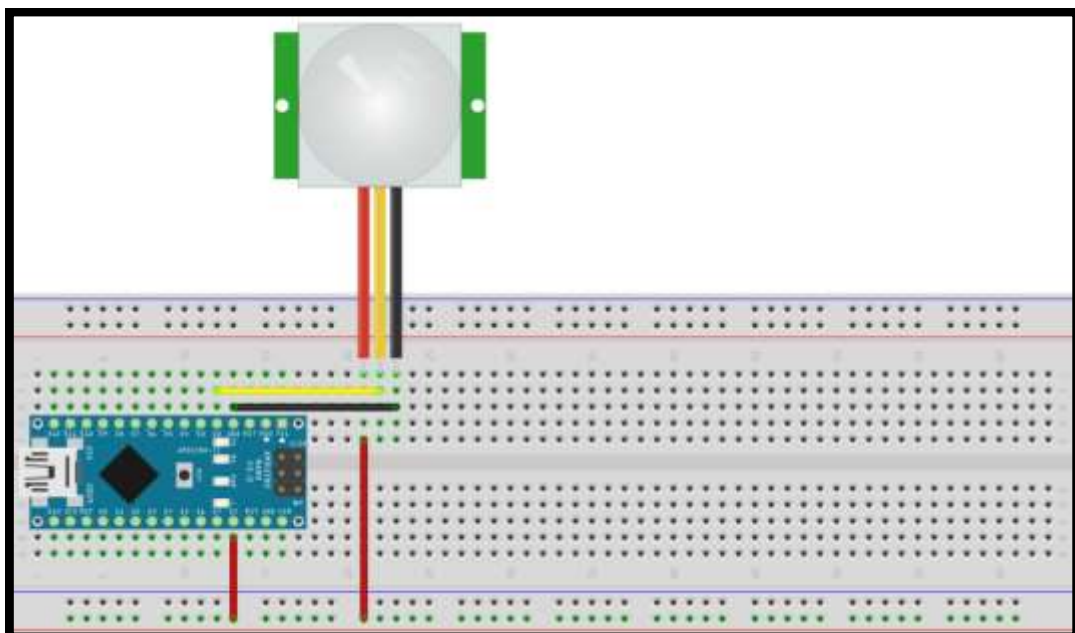
Sensor ini biasanya digunakan untuk pendeteksi gerakan, karena semua benda memancarkan energi radiasi maka sebuah gerakan akan terdeteksi ketika melewati sensor infra merah ini. Sensor PIR terdiri dari beberapa bagian, diantaranya yaitu :

- Lensa Fresnel
- Penyaring Infra Merah
- Sensor Pyroelektrik
- Penguat Amplifier
- Komparator

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. Sensor PIR
4. Kabel Jumper

b. LANGKAH PRAKTIKUM



Gambar 6.4. Rangkaian Arduino Nano dengan Sensor PIR

```

int calibrationTime = 30;
long unsigned int lowIn;
long unsigned int pause = 5000;
boolean lockLow = true;
boolean takeLowTime;

int pirPin = 2;
void setup(){
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  digitalWrite(pirPin, LOW);

  Serial.print("calibrating sensor ");
  for(int i = 0; i < calibrationTime; i++){
    Serial.print(".");
    delay(1000);
  }
  Serial.println(" done");
  Serial.println("SENSOR ACTIVE");
  delay(50);
}

void loop(){

  if(digitalRead(pirPin) == HIGH){
    if(lockLow){
      lockLow = false;
      Serial.println("---");
      Serial.print("motion detected at ");
      Serial.print(millis()/1000);
      Serial.println(" sec");
      delay(50);
    }
    takeLowTime = true;
  }

  if(digitalRead(pirPin) == LOW){
    if(takeLowTime){
      lowIn = millis();
      takeLowTime = false;
    }
    if(!lockLow && millis() - lowIn > pause){
      lockLow = true;
      Serial.print("motion ended at ");
      Serial.print((millis() - pause)/1000);
      Serial.println(" sec");
      delay(50);
    }
  }
}

```

Modul 6.3. Program Arduino dengan Sensor PIR

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.

3. Pasangkan kabel jumper arduino sesuai dengan pin yang ada di rangkaian.
4. Lalu pasang kabel jumper yang sudah terpasang di pin arduino ke sensor dengan hati hati.
5. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
6. Buka serial monitor, bergeraklah didepan sensor.
7. Amati apa yang terjadi.

TUGAS PRAKTIKUM

1. Buatlah program untuk HC-SR04 dan LED dengan kondisi jika jarak 10 – 50 cm maka LED akan hidup, buat rangkaiannya dan coba implementasikan !

MODUL 7

KOMUNIKASI NIRKABEL

DENGAN BLUETOOTH HC-05

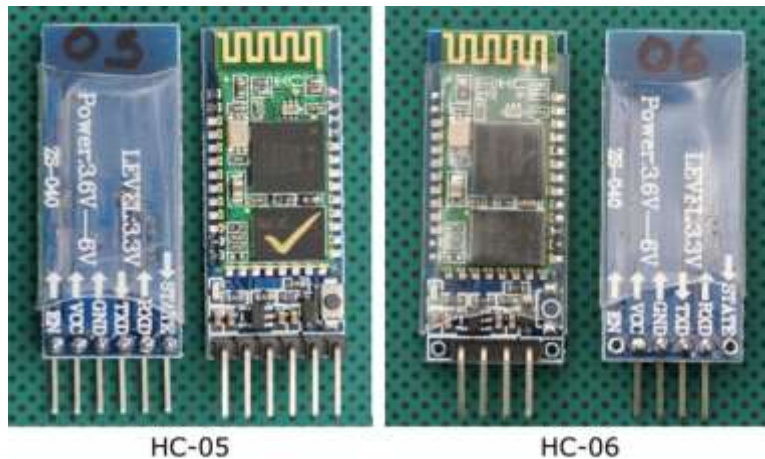
Bluetooth merupakan salah satu bentuk interkoneksi nirkabel yang sangat populer di kalangan pengguna smart phone. Seluruh smart phone modern memiliki modul bluetooth terintegrasi. Termasuk dalam Personal Area Networks, limitasi Bluetooth adalah range atau jangkauan penerimaan ataupun pengiriman data.

Pada praktikum ini Bluetooth (modul Bluetooth HC-05) akan digunakan sebagai media interkoneksi dan komunikasi nirkabel antara board Arduino dengan smart phone dan juga dengan komputer/PC. Selain itu integrasi antara modul Bluetooth HC-05 dengan modul sensor juga akan diimplementasikan.

HC-05 (zs-040) dan HC-06 (zs-040) menggunakan platform yang sama (bahkan memiliki cetak layar yang sama) tetapi memiliki beberapa perbedaan :

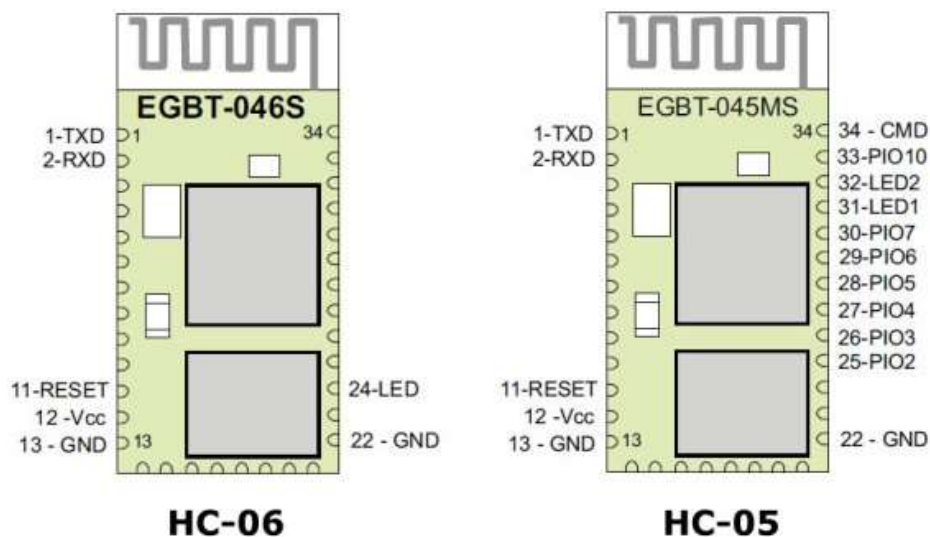
- HC-06 tidak memiliki tombol reset
- HC-06 hanya 4 pin header
- HC-06 tidak memiliki pin 31-34 yang terhubung

Keduanya juga memiliki firmware yang berbeda. HC-05 dapat berfungsi sebagai master atau slave. HC-06 hanya sebagai slave. Sehingga HC-05 dapat membangun koneksi ke perangkat lain dan HC-06 hanya dapat menerima koneksi dari perangkat lain.



Gambar 7.1. Modul Bluetooth HC-05 dan HC-06

Board pada HC 05 dan Hc-06 dibuat berdasarkan pada modul Bluetooth EGBT-045MS / EGBT-046S yang dikemas dengan firmware SPP untuk komunikasi UART.



Gambar 7.2. Modul HC-06 dengan EGBT-046S dan HC-05 dengan EGBT-045MS

Modul EGBT-045MS dan EGBT-046S memiliki perangkat keras yang sama dan satu-satunya perbedaan adalah firmware. Firmware yang berbeda tidak mengubah pin out. Boards pada zs-040 telah dilengkapi regulator tegangan 3.3V sehingga dapat menerima VCC 3.6V hingga 6V pada pin / konektor VCC utama sedangkan, pin RX dan TX masih memerlukan tegangan 3.3V.

Arduino akan menerima 3.3V sebagai sinyal HIGH sehingga pin TX (out) HC-05/06 dapat langsung terhubung ke Arduino 5V. Pin HC-05/06 RX (in) tidak dapat

menerima 5V dan seharusnya tidak terhubung langsung ke Arduino. Jika terjadi sambungan langsung akan bekerja dalam jangka pendek tetapi merusak modul bluetooth dan akhirnya akan menyebabkan pin RX rusak. Voltase 5V dari Arduino dapat dikurangi dengan menggunakan pembagi tegangan yang terbuat dari 2 resistor yakni resistor 1K Ohm dan resistor 2K Ohm

Pada HC 05 memiliki pin EN dan STATE, sedangkan HC 06 tidak dilengkapi pin tersebut. Tombol (Push Button Switch) menghubungkan VCC (3.3V) ke pin 34 dan digunakan untuk mengatur EGBT-045MS / HC-05 ke mode AT. Ada 2 mode AT, dapat disebut sebagai "mode mini" dan "mode penuh". Dengan menekan tombol beberapa detik maka akan menempatkan modul ke mode "mini" sehingga tidak semua perintah berfungsi. Dengan mengkondisikan switch tetap tertutup, maka modul akan masuk ke mode AT penuh sehingga semua perintah AT bekerja. Melepaskan switch mengembalikan pin 34 ke kondisi LOW. Sebagian besar modul HC-06 tidak memiliki tombol (ruang kosong) dan membawa pin 34 High di HC-06 tidak menyebabkan apapun.

PAIRING MODUL HC-05 DAN SMART PHONE BERBASIS ANDROID

ALAT DAN BAHAN

1. Arduino NANO
2. Modul Bluetooth HC-05
3. Breadboard
4. Kabel jumper
5. Resistor 1K Ohm dan 2K Ohm

RANGKAIAN DAN LANGKAH PRAKTIKUM

1. Install aplikasi **Bluetooth Terminal HC-05** (developer mightyIT, <https://play.google.com/store/apps/details?id=project.bluetoothterminal&hl=en>) pada smartphone berbasis Android untuk melakukan uji coba konektivitas.

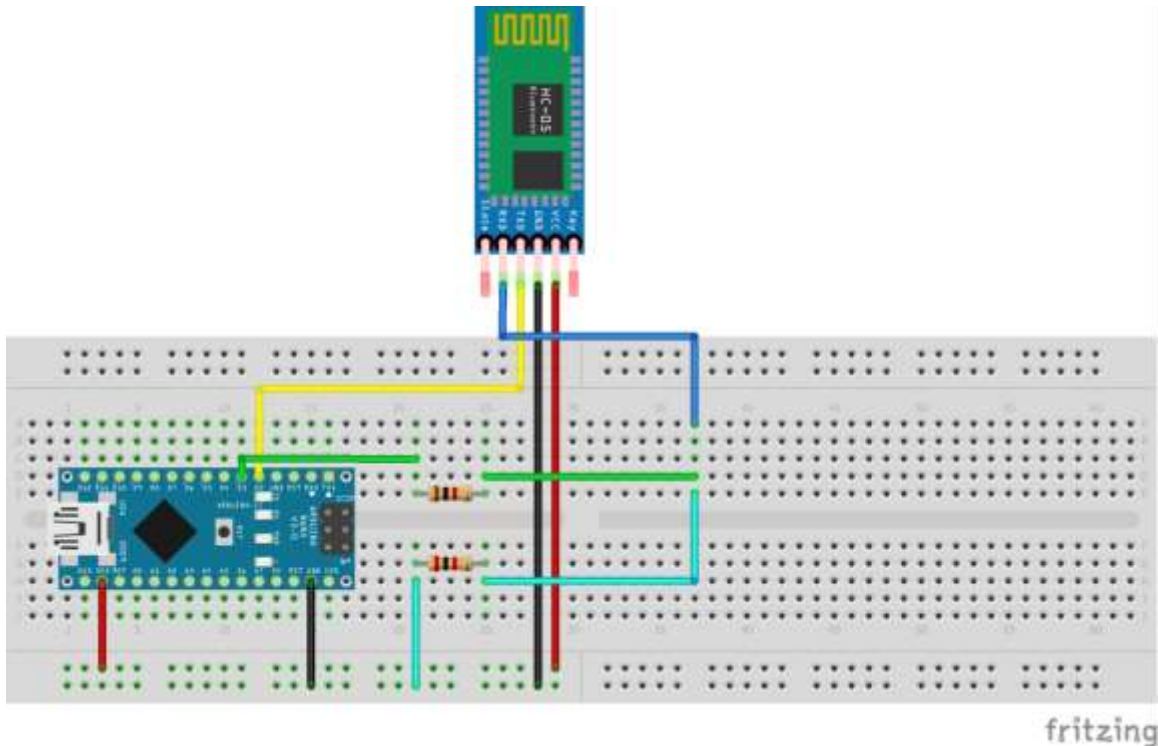
2. Buat rangkaian seperti gambar 7.1. Perhatikan baik-baik ukuran resistor, salah memasang resistor bisa berakibat terbakarnya modul HC-05. Hubungkan rangkaian dengan komputer.
3. Tulis kode berikut ke Arduino IDE, verify/compile kemudian upload ke board Arduino.

```
#include <SoftwareSerial.h>
SoftwareSerial BTserial(2, 3);

char c = ' ';

void setup(){
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  BTserial.begin(9600);
}

void loop(){
  BTserial.println("Bluetooth Test");
  Serial.println("Bluetooth Test");
  delay(1000);
}
```



Gambar 7.3. Rangkaian Bluetooth HC-05

4. Setelah program selesai di-upload, buka aplikasi Bluetooth Terminal HC-05 di smart phone anda, berikan izin ketika aplikasi meminta izin untuk menyalakan modul Bluetooth dan meminta akses lokasi/GPS di HP anda. Kemudian lakukan scan dengan memilih opsi SCAN di pojok kanan atas.
5. Pilih modul Bluetooth yang muncul (umumnya bernama HC-05 atau HC-06), kemudian masukkan angka 1234 atau 0000 sebagai default password untuk melakukan pairing antara smart phone anda dan modul Bluetooth.
6. Setelah pairing berhasil dilakukan, buka terminal Bluetooth HC-05. Selain itu buka juga Serial Monitor pada komputer (sinkronkan nilai BAUD rate).
7. Amati apa yang terjadi.

KOMUNIKASI SERIAL ANTARA ARDUINO DAN SMART PHONE DENGAN BLUETOOTH HC-05

ALAT DAN BAHAN

1. Arduino NANO
2. Modul Bluetooth HC-05
3. Breadboard
4. Kabel jumper
5. Resistor 1K Ohm dan 2K Ohm

RANGKAIAN DAN LANGKAH PRAKTIKUM

1. Rangkaian masih menggunakan rangkaian percobaan sebelumnya.
2. Tulis kode berikut ke Arduino IDE, verify dan compile, lalu upload ke board Arduino Nano.

```
#include <SoftwareSerial.h>
SoftwareSerial BTserial(2, 3);
char c = ' ';

void setup() {
  Serial.begin(9600);
  Serial.println("Arduino is ready");
  BTserial.begin(9600);
}
```

```
void loop() {  
    if (BTserial.available()) {  
        c = BTserial.read();  
        Serial.write(c);  
    }  
  
    if (Serial.available()) {  
        c = Serial.read();  
        BTserial.write(c);  
    }  
}
```

3. Buka aplikasi smart phone Bluetooth Terminal HC-05 kemudian lakukan pairing bila diperlukan (seharusnya tidak, modul HC-05 sudah terdaftar sebagai paired device).
4. Buka terminal HC-05 di smart phone, dan buka Serial Monitor (Ctrl + Shift + M) di Arduino IDE.
5. Tuliskan pesan singkat di terminal HC-05 (smart phone), amati apa yang terjadi !
6. Tuliskan pesan singkat di Serial Monitor pada Arduino IDE, amati apa yang terjadi !

INTEGRASI SENSOR DHT11 DENGAN BLUETOOTH HC-05

ALAT DAN BAHAN

1. Arduino NANO
2. Modul Bluetooth HC-05
3. Breadboard
4. Kabel jumper
5. Resistor 1K Ohm dan 2K Ohm
6. Sensor DHT11

RANGKAIAN DAN LANGKAH PRAKTIKUM

1. Tambahkan sensor DHT11 ke dalam rangkaian sebelumnya seperti pada gambar 7.2.

2. Tulis kode berikut ke Arduino IDE, verify dan compile, lalu upload ke board Arduino Nano.

```
#include <SoftwareSerial.h>
#include <DHT.h>
#define DHTTYPE DHT11

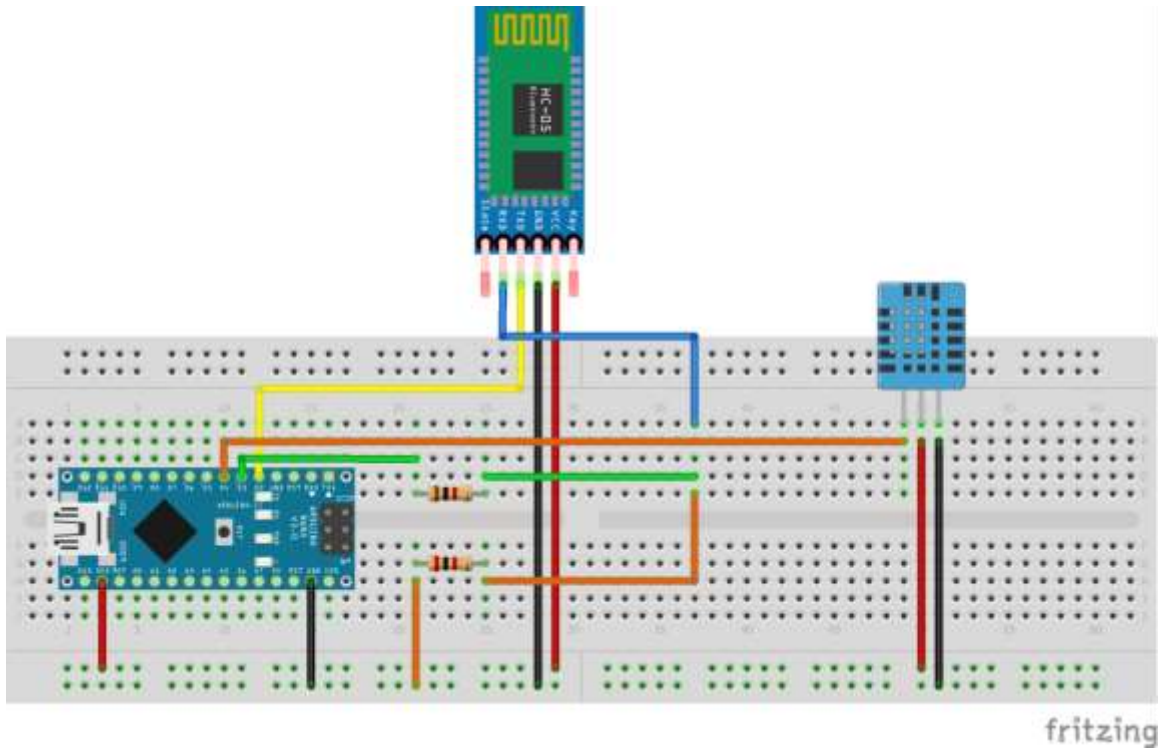
SoftwareSerial BTserial(2, 3); // RX | TX
char c = ' ';
const int DHTpin = 4;
DHT dht(DHTpin, DHTTYPE);

void setup(){
    Serial.begin(9600);
    Serial.println("Arduino is ready");
    dht.begin();
    BTserial.begin(9600);
}

void loop(){
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    BTserial.println(h);
    BTserial.println(t);
    delay(5000);
}
```

3. Buka aplikasi smart phone Bluetooth Terminal HC-05 kemudian lakukan pairing bila diperlukan (seharusnya tidak, modul HC-05 sudah terdaftar sebagai paired device).



Gambar 7.4. Rangkaian Integrasi DHT11 dan HC-05

4. Buka terminal HC-05 di smart phone kemudian tunggu hingga selama waktu delay yang di-set di program (5000 ms atau 5 detik), amati apa yang terjadi pada terminal HC-05 di smart phone anda !

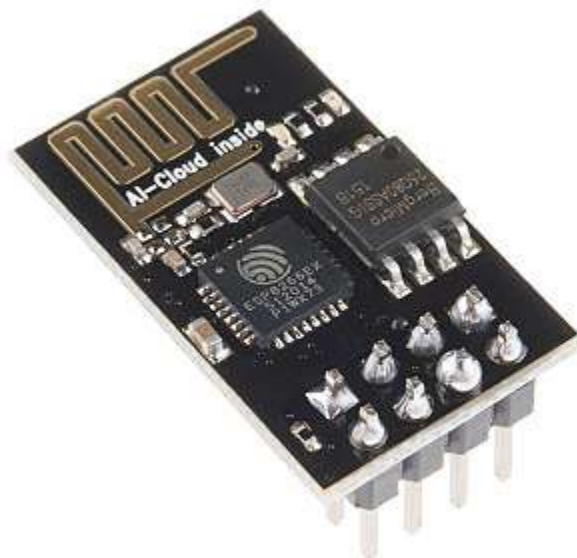
MODUL 8

PENGENALAN ESP8266 DENGAN BLYNK

ESP8266 merupakan low-cost Wi-Fi microchip yang diproduksi oleh Espressif Systems (Shanghai, Cina). ESP8266 merupakan sebuah modul terintegrasi yang memiliki mikrokontroler terintegrasi sekaligus TCP/IP stack yang lengkap dengan standar Wi-Fi 802.11 b/g/n. ESP8266 menjadi populer di kalangan maker setelah digunakan oleh perusahaan Ai-Thinker pada modul terintegrasi ESP-01 dengan harga terjangkau pada tahun 2014.



Gambar 8.1. Mikrochip ESP8266EX (Wikipedia)

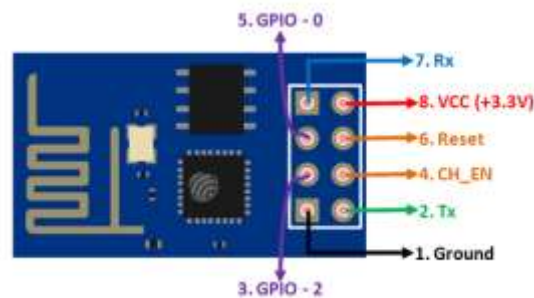


Gambar 8.2. Modul ESP-01 produksi Ai-Thinker (Wikipedia)

Spesifikasi perangkat keras dari ESP8266 adalah sebagai berikut.

- Prosesor : L106 berbasis Tensilica Xtensa Diamond Standard 106Micro, 32-bit RISC, dengan clock 80 Mhz,
- Memory : 32 KB RAM untuk instruksi, 32 KB RAM untuk cache instruksi, 80 KB RAM untuk user data, dan 16 KB untuk ETS system-data RAM,
- QSPI flash eksternal, didukung hingga 16 MB (umum dijumpai pada modul bawaan dengan ukuran 512 KB – 4 MB),
- IEEE 802.11 b/g/n Wi-Fi dengan autentikasi WEP atau WPA/WPA2,
- GPIO berjumlah 16 pin,
- SPI, I2C, I2S, dan UART,
- 10-bit ADC.

Sedangkan pinout untuk modul ESP-01 disajikan dalam gambar berikut.



Gambar 8.3. Pinout ESP-01 (components101.com)

Untuk pemrograman ESP8266, Espressif Systems menyediakan 2 macam SDK, SDK yang berbasis FreeRTOS dan yang berbasis callbacks. Komunitas maker sendiri terus mengembangkan berbagai macam SDK dan firmware untuk memprogram ESP8266, salah satunya adalah yang digunakan pada Arduino (bisa diunduh dari repository resmi Arduino melalui Arduino IDE atau melalui GitHub).

Salah satu kelemahan integrasi ESP8266 dengan Arduino (terutama pada Uno dan Nano) adalah permasalahan suplai daya. Arus yang dapat disuplai melalui board Arduino terkadang kurang untuk memberi daya operasi pada modul ESP8266, terutama ketika modul ESP8266 beroperasi pada mode full load (misal : mengirim data melalui modul Wi-Fi dalam jangka waktu yang cukup lama). Hal ini diatasi dengan menggunakan catu daya eksternal (3.3v) atau dengan menggunakan modul ESP8266

yang sudah diintegrasikan dengan catu daya independen melalui USB-to-UART bridge yang dilengkapi dengan regulator voltase 3.3v independen. WEMOS dan NodeMCU merupakan 2 board dengan modul ESP8266 terintegrasi dengan USB-to-UART dan regulator voltase independen yang populer di kalangan maker.



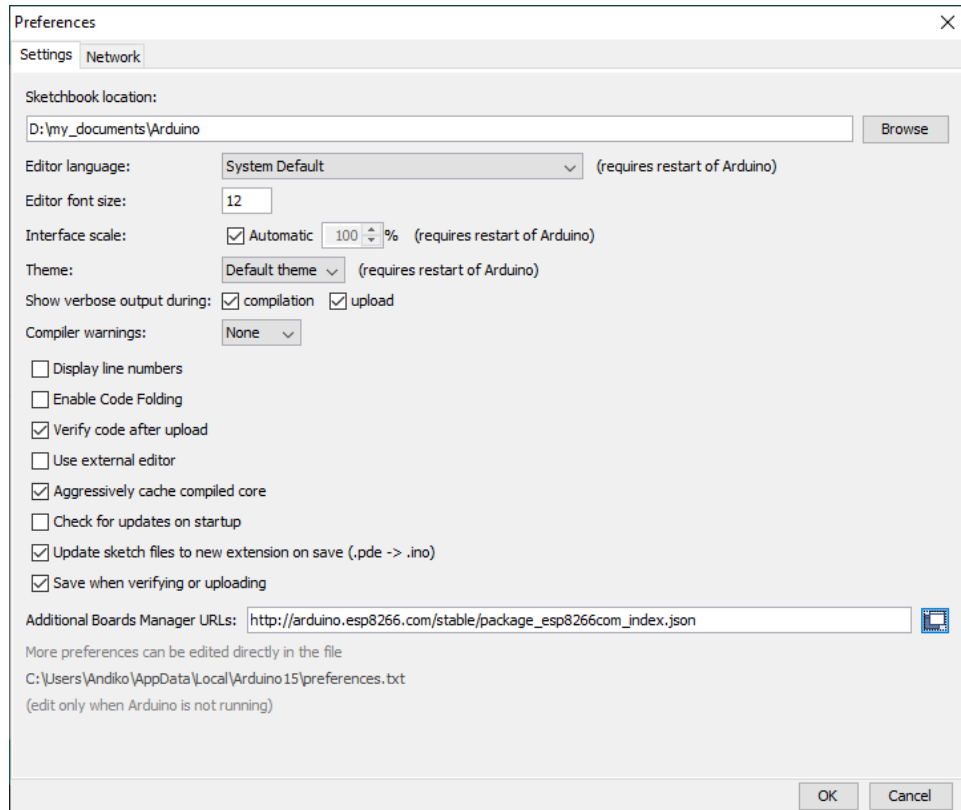
Gambar 8.4. WEMOS D1 Mini (kiri) dan NodeMCU (kanan)

PENGENALAN EP8266 DENGAN BLYNK IoT PLATFORM

INSTALASI BOARD MANAGER ESP8266 DI ARDUINO IDE

Supaya Arduino IDE dapat digunakan untuk memprogram ESP8266, perlu dilakukan instalasi board manager dengan menggunakan langkah-langkah instalasi library dan board manager pada BAB/Modul 2.

Pastikan komputer terhubung dengan internet, kemudian di Arduino IDE masuk ke menu Preferences, pada field isian Additional Boards Manager URLs isikan URL berikut ini : http://arduino.esp8266.com/stable/package_esp8266com_index.json. Klik tombol “Ok” dan tunggu hingga proses download board manager selesai.

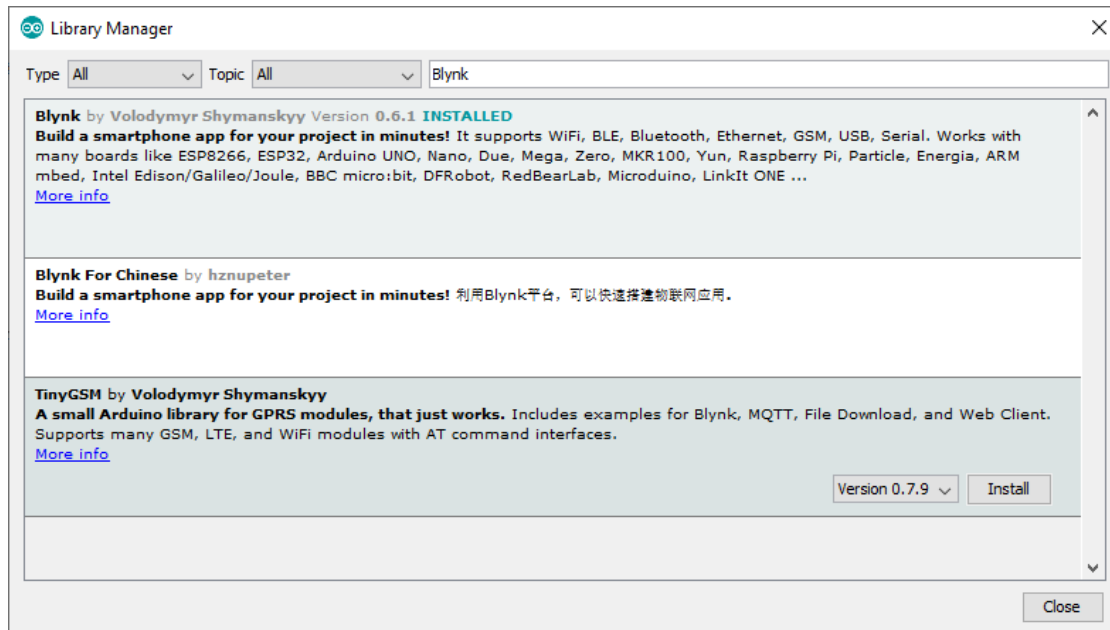


Gambar 8.5. Jendela Preferences untuk menambahkan Board Manager

INSTALASI LIBRARY BLYNK

BLYNK merupakan sebuah platform IoT yang berbasis aplikasi mobile dan dapat digunakan dengan gratis maupun membayar. Esensi dari BLYNK adalah menghubungkan berbagai macam "thing" berupa device dan sensor dengan smart phone melalui platform IoT mereka.

Untuk dapat menggunakan platform Blynk, user harus mendownload aplikasi BLYNK melalui Play Store (Google) atau Apps Store (Apple) dan melakukan registrasi, search BLYNK dengan kata kunci "blynk iot arduino". Dari sisi Arduino IDE, kita perlu melakukan instalasi library Blynk dengan cara instalasi library pada Modul/Bab 2, seach library dengan kata kunci Blynk.



Gambar 8.6. Instalasi Library BLYNK

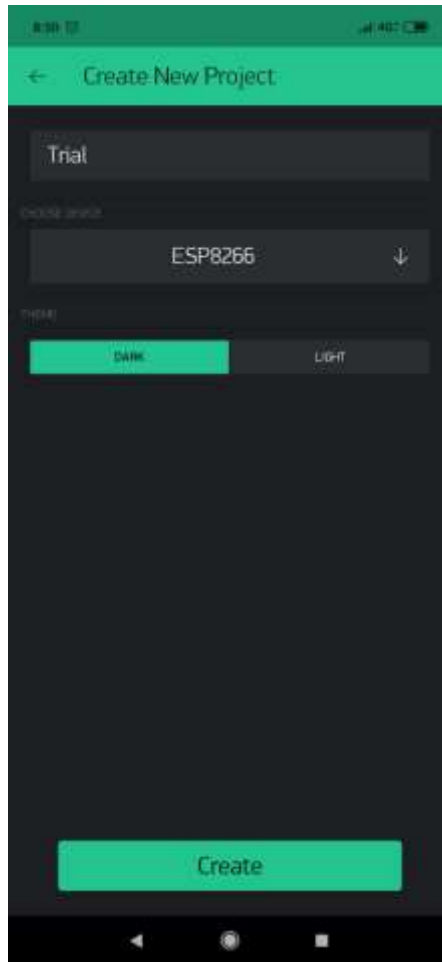
IMPLEMENTASI SEDERHANA ESP8266 DENGAN BLYNK

ALAT DAN BAHAN

1. Modul ESP8266 Wemos Lolin V3
2. Kabel jumper
3. Breadboard
4. LED
5. Resistor 220 Ohm
6. Smart phone
7. Jaringan WiFi atau internet

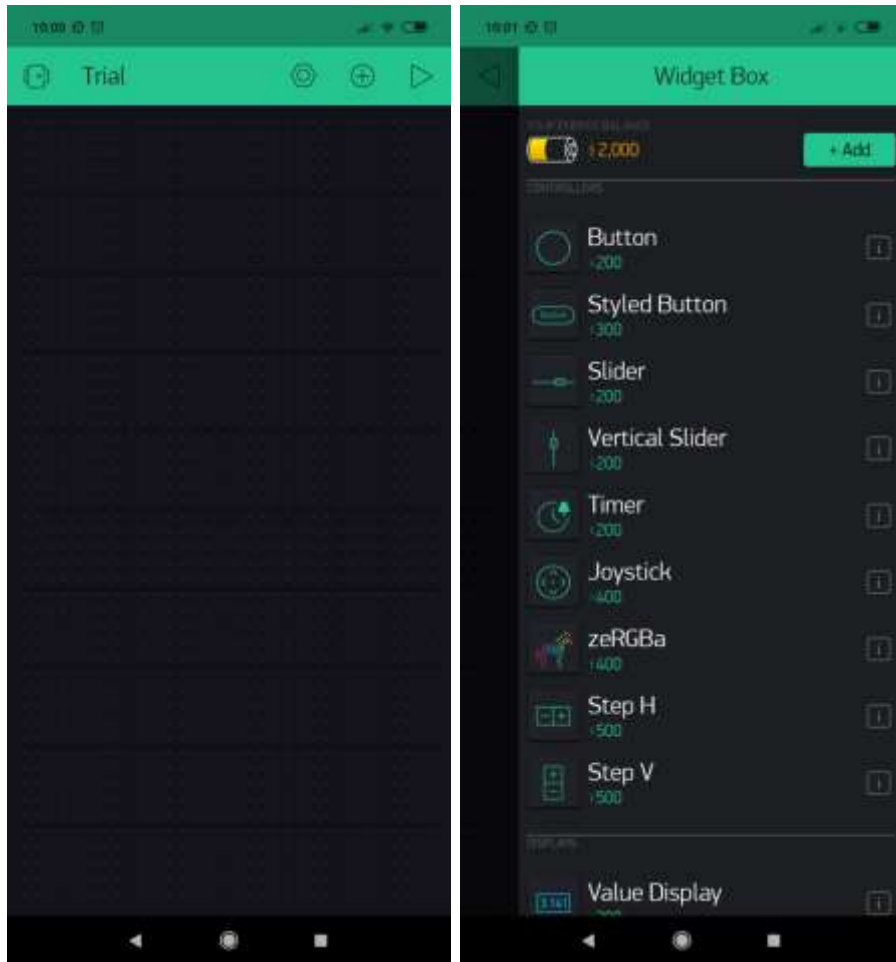
RANGKAIAN DAN LANGKAH PRAKTIKUM

1. Buatlah rangkaian sesuai dengan gambar 8.7.
2. Buka aplikasi BLYNK di smart phone anda, login apabila sudah memiliki account atau registrasi account baru apabila belum ada account.
3. Setelah login, pilih menu New Project. Beri nama project pada isian Project Name, dan pilih ESP8266 pada menu drop down CHOOSE DEVICE.



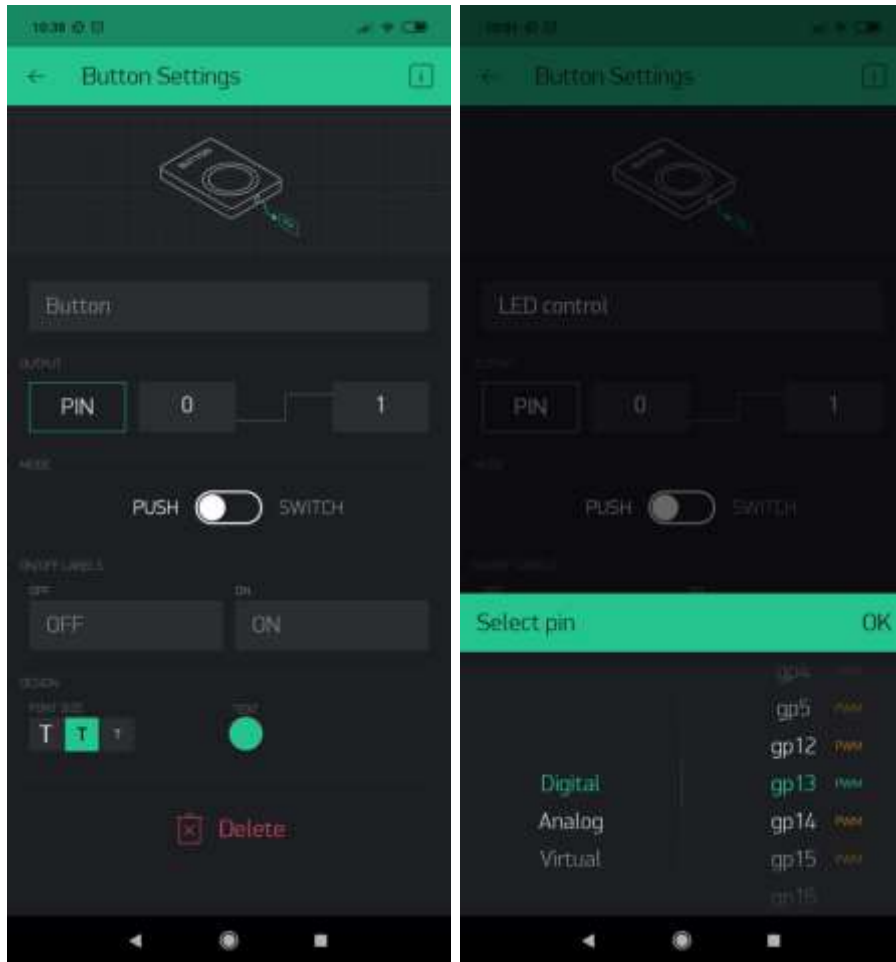
Gambar 8.7. Tampilan New Project pada Blynk

4. Kemudian BLYNK akan mengirimkan kode otorisasi (authorization key) ke email anda (yang digunakan untuk register). Kode otorisasi juga bisa anda akses pada menu Settings dari tiap Project (icon baut).



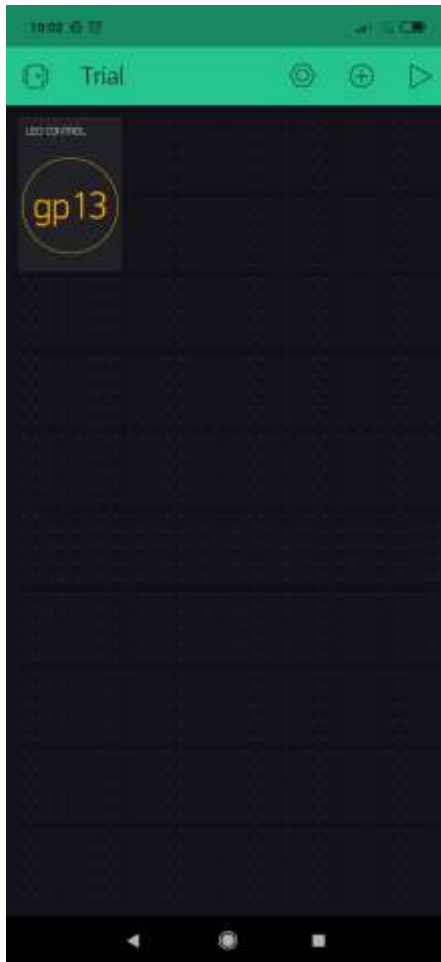
Gambar 8.8. Interface Project (kiri) dan Widget Box (kanan)

5. Di halaman Project akan tampil interface seperti pada gambar 8.8 (kiri). Pilih menu Settings di kanan atas untuk menyalin atau mengirim ulang kode otorisasi. Pilih menu Add Widgets maka akan muncul side menu Widget Box (gambar 8.8 kanan). Pilih Button dari side menu Widget Box, kemudian akan muncul tombol pada halaman Project.



Gambar 8.9. Menu Button settings (kiri) dan pemilihan Output PIN (kanan)

6. Pilih Button yang baru muncul, kemudian atur konfigurasi Button nya (Gambar 8.9 kiri). Beri nama button, kemudian pada bagian Output (PIN) pilih pin Digital dan gp13 (Gambar 8.9 kanan), biarkan angka 0 dan 1 di samping opsi Output PIN. Mode switch bisa dipilih PUSH (akan aktif selama tombol dipencet) atau SWITCH (toggle). Setelah button selesai dikonfigurasi, akan muncul tampilan seperti gambar 8.10. Selanjutnya kita akan melakukan konfigurasi pada Arduino IDE.



Gambar 8.10. Tampilan BLYNK setelah konfigurasi Button

7. Tulis dan upload kode berikut ke Arduino IDE. Perhatikan baik-baik **bagian kode yang dicetak tebal**.

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "<b><ganti dengan auth key anda></b>";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "<b><ganti dengan nama AP></b>";
char pass[] = "<b><ganti dengan password AP></b>";

void setup()
{
  // Debug console
  Serial.begin(9600);
```

```
Blynk.begin(auth, ssid, pass);  
}  
  
void loop()  
{  
  Blynk.run();  
}
```

8. Untuk kode otorisasi (authorization key/token) isikan dengan kode otorisasi yang dikirimkan BLYNK ke email anda atau copy dan paste dari menu Settings (baut) pada jendela Project.
9. Setelah kode program Arduino IDE di-upload, kita akan kembali ke smart phone (aplikasi BLYNK).
10. Dari jendela Project yang sudah memuat button, pilih tanda Play di bagian atas paling kanan. Setelah itu coba pencet button yang anda buat tadi. Amati apa yang terjadi, cermati apabila terjadi error!

MODUL 9

ESP8266 STANDALONE WEB SERVER

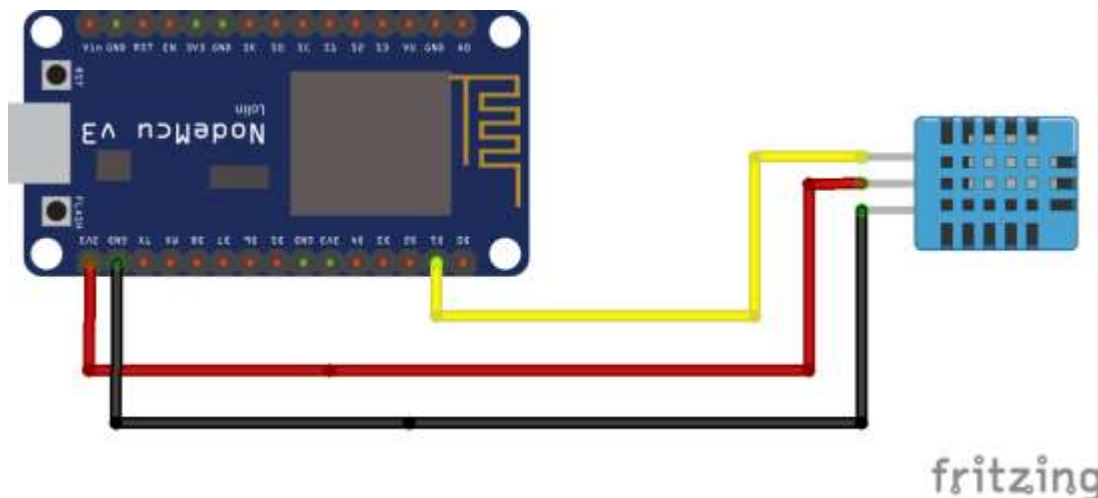
ESP8266 memiliki seluruh bagian/layer dari TCP/IP, sehingga secara independen ESP8266 mampu menjalankan fungsi sebagai application layer sekaligus transport layer. Kita dapat memprogram ESP8266 untuk menjalankan sebuah aplikasi web sederhana (dalam contoh ini adalah sebuah web server) sekaligus menjadikan ESP8266 sebagai media komunikasi antara sensor (things) dengan layer network.

Web server sederhana yang dibuat akan digunakan untuk mengambil data sensor DHT11 (temperatur dan kelembaban udara) untuk kemudian disajikan dalam sebuah tampilan web yang dapat diakses melalui jaringan nirkabel (WiFi).

ALAT DAN BAHAN

1. Modul ESP8266 Wemos Lolin (V3)
2. Kabel jumper (male-female x 4)
3. Breadboard
4. Sensor DHT11
5. Jaringan WiFi dan atau internet

RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 9.1. Rangkaian ESP8266 Standalone Web Server

Langkah-langkah praktikum dijabarkan sebagai berikut.

1. Buat rangkaian seperti pada gambar di atas. Kemudian tulis kode berikut pada Arduino IDE. Perhatikan baik-baik **bagian kode yang dicetak tebal**.

```
#include <ESP8266WiFi.h>
#include "DHT.h"
#define DHTTYPE DHT11

const char* ssid = "<b><ganti dengan nama AP yang digunakan></b>";
const char* password = "<b><ganti dengan password AP yang digunakan (bila ada)></b>";

WiFiServer server(80);

const int DHTPin = 5;
DHT dht(DHTPin, DHTTYPE);

static char celsiusTemp[7];
static char fahrenheitTemp[7];
static char humidityTemp[7];

void setup() {
  Serial.begin(115200);
  delay(10);
  dht.begin();

  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  server.begin();
  Serial.println("Web server running. Waiting for the ESP IP...");
  delay(1000);
  Serial.println(WiFi.localIP());
}

void loop() {
  WiFiClient client = server.available();

  if (client) {
    Serial.println("New client");
    boolean blank_line = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
```

```

        if (c == '\n' && blank_line) {
            float h = dht.readHumidity();
            float t = dht.readTemperature();
            float f = dht.readTemperature(true);
            if (isnan(h) || isnan(t) || isnan(f)) {
                Serial.println("Failed to read from DHT sensor!");
                strcpy(celsiusTemp, "Failed");
                strcpy(fahrenheitTemp, "Failed");
                strcpy(humidityTemp, "Failed");
            }
            else{
                float hic = dht.computeHeatIndex(t, h, false);
                dtostrf(hic, 6, 2, celsiusTemp);
                float hif = dht.computeHeatIndex(f, h);
                dtostrf(hif, 6, 2, fahrenheitTemp);
                dtostrf(h, 6, 2, humidityTemp);
            }
            client.println("HTTP/1.1 200 OK");
            client.println("Content-Type: text/html");
            client.println("Connection: close");
            client.println();
            client.println("<!DOCTYPE HTML>");
            client.println("<html>");
            client.println("<head></head><body><h1>ESP8266 -
Temperature and Humidity</h1><h3>Temperature in Celsius: ");
            client.println(celsiusTemp);
            client.println("*C</h3><h3>Temperature in Fahrenheit: ");
            client.println(fahrenheitTemp);
            client.println("*F</h3><h3>Humidity: ");
            client.println(humidityTemp);
            client.println("%</h3><h3>");
            client.println("</body></html>");
            break;
        }
        if (c == '\n') {
            blank_line = true;
        }
        else if (c != '\r') {
            blank_line = false;
        }
    }
}
delay(1);
client.stop();
Serial.println("Client disconnected.");
}
}

```

2. Verify/compile dan bila tidak ada error, buka Serial Monitor, kemudian upload kode tersebut ke modul ESP8266.

3. Tujuan membuka Serial Monitor sebelum melakukan upload program ke modul ESP8266 adalah supaya kita bisa langsung mendapatkan informasi alamat IP dari modul ESP8266 (tanpa harus melakukan reset pada modul).
4. Akses web server sederhana dengan cara membuka alamat IP dari modul ESP8266 di browser yang terhubung pada AP yang sama.
5. Amati hasilnya dan catat serta cermati apabila ditemukan error.

TUGAS PRAKTIKUM

1. Buatlah Standalone Web Server dengan ES8266 dengan menggunakan sensor lain selain DHT11!

MODUL 10

DATA LOGGING DENGAN ESP8266 DAN MySQL

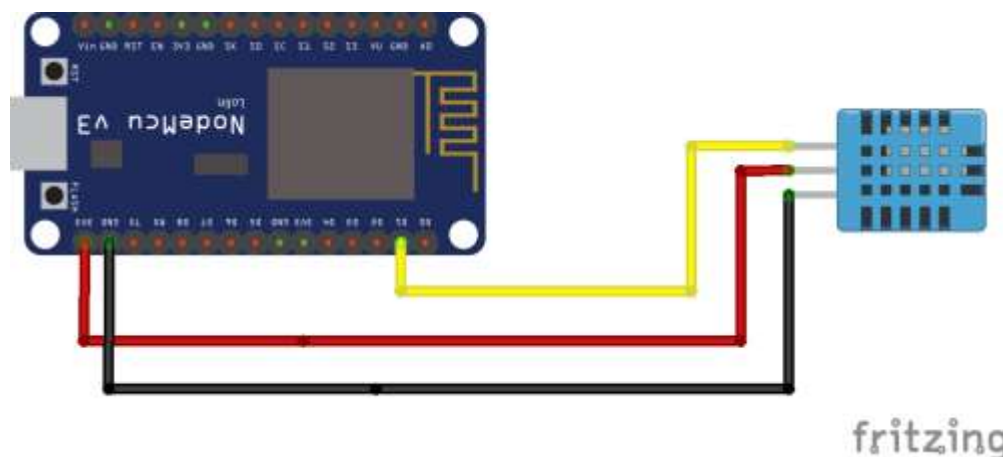
ESP8266 (NodeMCU) memiliki kemampuan untuk mengirimkan pesan HTTP (HTTP request) seperti POST dan GET. Hal ini memungkinkan untuk mendesain sebuah sistem data logging yang berfungsi untuk menyimpan data hasil bacaan sensor yang terhubung ke ESP8266 ke sebuah basis data melalui jaringan internet atau WLAN.

Pada praktikum ini akan dibuat sebuah implementasi data logging dengan menggunakan modul ESP8266 (NodeMCU V3) dan database MySQL dengan menggunakan local server (localhost). Web server akan menyimpan data bacaan sensor DHT11 berupa data temperatur dan kelembaban udara serta timestamp.

ALAT DAN BAHAN

1. Modul ESP8266 Wemos Lolin (V3)
2. Kabel jumper (male-female x 3)
3. Breadboard
4. Sensor DHT11
5. Jaringan WiFi

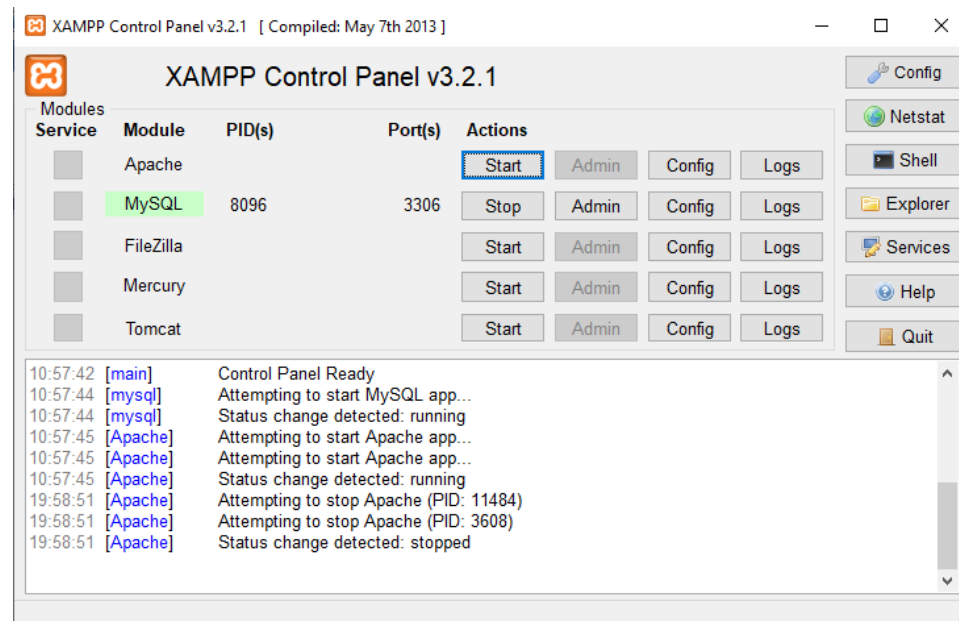
RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 10.1. Rangkaian ESP8266 dan DHT11

Langkah-langkah praktikum dijabarkan sebagai berikut.

1. Persiapkan local server (pada komputer/laptop) dengan melakukan instalasi web server dengan menggunakan XAMPP. Pastikan service MySQL server menyala pada control panel XAMPP seperti pada gambar.



Gambar 10.2. Jendela XAMPP Control Panel

2. Buka PHPMyAdmin dan buat database baru dengan nama **esp8266** (UTF-8). Kemudian buat tabel baru dengan nama **dht11**. Gunakan syntax SQL berikut untuk membuat tabel **dht11**.

```
CREATE TABLE dht11 (
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  sensor VARCHAR(30) NOT NULL,
  temp VARCHAR(10),
  hum VARCHAR(10),
  reading_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
  CURRENT_TIMESTAMP
)
```

3. Sebagai medium untuk melakukan proses INSERT pada database, buatlah sebuah file dengan nama **dht11.php** sebagai berikut.

```
<?php

$servername = "localhost";

$dbname = "esp8266";
$username = "root";
$password = "";
```

```

$api_key_value = "123456789";

$api_key= $sensor = $temp = $hum = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $api_key = test_input($_POST["api_key"]);
    if($api_key == $api_key_value) {
        $sensor = test_input($_POST["sensor"]);
        $temp = test_input($_POST["temp"]);
        $hum = test_input($_POST["hum"]);

        $conn = new mysqli($servername, $username, $password,
$dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Koneksi Gagal: " . $conn->connect_error);
        }

        $sql = "INSERT INTO dht11 (sensor, temp, hum)
VALUES ('" . $sensor . "', '" . $temp . "', '" . $hum .
'')";

        if ($conn->query($sql) === TRUE) {
            echo "Record baru berhasil dibuat ";
        }
        else {
            echo "Error: " . $sql . "<br>" . $conn->error;
        }

        $conn->close();
    }
    else {
        echo "Verifikasi API Key gagal.";
    }
}
else {
    echo "Tidak ada data dikirim melalui HTTP POST.";
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

```

4. Sebagai medium untuk melakukan proses SELECT pada database sekaligus menampilkan record pada database, buatlah sebuah file dengan nama **dht11_view.php** sebagai berikut.

```
<!DOCTYPE html>
<html><body>
<?php

$servername = "localhost";

$dbname = "esp8266";
$username = "root";
$password = "";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Koneksi gagal : " . $conn->connect_error);
}

$sql = "SELECT id, sensor, temp, hum, reading_time FROM dht11";

echo '<table cellpadding="5" cellspacing="5">
    <tr>
        <td>ID</td>
        <td>Sensor</td>
        <td>Temperatur</td>
        <td>Kelembaban Udara</td>
        <td>Waktu</td>
    </tr>';

if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_sensor = $row["sensor"];
        $row_temp = $row["temp"];
        $row_hum = $row["hum"];
        $row_reading_time = $row["reading_time"];

        echo '<tr>
            <td>' . $row_id . '</td>
            <td>' . $row_sensor . '</td>
            <td>' . $row_temp . '</td>
            <td>' . $row_hum . '</td>
            <td>' . $row_reading_time . '</td>
        </tr>';
    }
    $result->free();
}

$conn->close();
```

```
?>
</table>
</body>
</html>
```

5. Simpan kedua file PHP tersebut ke dalam folder **dht11** kemudian letakkan folder tersebut ke dalam folder **htdocs** di lokasi instalasi XAMPP.
6. Buat rangkaian sesuai dengan diagram pada gambar ??.
7. Buat program pada Arduino IDE dengan kode sebagai berikut. Perhatikan baik-baik **bagian kode yang dicetak tebal**.

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include "DHT.h"
#include <Adafruit_Sensor.h>

#define DHTTYPE DHT11
const char* ssid      = "<b><ganti dengan nama AP yang digunakan></b>";
const char* password  = "<b><ganti dengan password AP bila ada></b> ";

const char*  serverName  =  "http://<b><ganti dengan alamat IP localhost></b>/dht11/dht11.php";

String apiKeyValue = "123456789";

String sensorName = "DHT11";

const int DHTPin = 5;
DHT dht(DHTPin, DHTTYPE);

static char celsiusTemp[7];
static char fahrenheitTemp[7];
static char humidityTemp[7];

void setup() {
  Serial.begin(115200);
  delay(10);

  dht.begin();
  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
}
```

```

    Serial.print("Terhubung dengan jaringan WiFi dengan alamat IP:
");
    Serial.println(WiFi.localIP());
}

void loop() {
    if(WiFi.status()== WL_CONNECTED){
        HTTPClient http;

        http.begin(serverName);

        http.addHeader("Content-Type","application/x-www-form-
urlencoded");

        float h = dht.readHumidity();
        float t = dht.readTemperature();
        String httpRequestData = "api_key=" + apiKeyValue +
"&sensor=" + sensorName + "&temp=" + String(t) +
"&hum=" + String(h) + "";
        Serial.print("httpRequestData: ");
        Serial.println(httpRequestData);

        int httpResponseCode = http.POST(httpRequestData);

        if (httpResponseCode>0) {
            Serial.print("HTTP Response code: ");
            Serial.println(httpResponseCode);
        }
        else {
            Serial.print("Error code: ");
            Serial.println(httpResponseCode);
        }
        http.end();
    }
    else {
        Serial.println("WiFi Terputus");
    }
    delay(10000);
}

```

8. Compile/verify kode tersebut, bila tidak diketemukan error, upload program ke modul ESP8266.
9. Amati data yang berhasil direkam ke dalam database dengan mengakses file **dht11_view.php** melalui browser (alamat **http://localhost/dht11/dht11_view.php**, apabila mengakses dari komputer/device lain ganti **localhost** dengan alamat IP web server).

10. Amati jalannya program secara keseluruhan, catat dan perhatikan baik-baik apabila ada error yang muncul.

TUGAS PRAKTIKUM

1. Tambahkan sebuah sensor