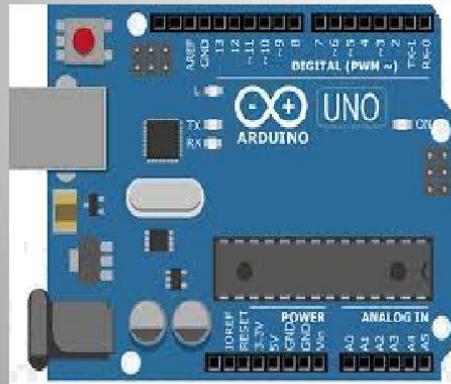


ARDUINO DAN SENSOR

pada project
Arduino DIY



Oleh : Dani Sasmoko ST.,M.Eng



Arduino dan Sensor Pada Project Arduino DIY

**DIGUNAKAN UNTUK MATAKULIAH
MIKROKONTROLER 2**

Oleh: Dani Sasmoko,S.T.,M.Eng.



YAYASAN PRIMA AGUS TEKNIK

Arduino dan Sensor pada Project Arduino DIY

Penulis:

Dani Sasmoko, S.T., M.Eng.

ISBN : 978-623-6141-37-3 (PDF)

Editor:

Indra Ava Dianta, S.Kom., M.T

Penyunting :

Danang, S.Kom., M.T

Desain Sampul dan Tata Letak :

Nuris Dwi Setiawan, S.Kom., M.T

Penerbit :

Yayasan Prima Agus Teknik

Redaksi: Jln Majapahit No 605 Semarang

Tlpn. (024) 6723456

Fax . 024-6710144

Email: penerbit_ypat@stekom.ac.id

Distributor Tunggal:

UNIVERSITAS STEKOM

Jln Majapahit No 605 Semarang

Tlpn. (024) 6723456

Fax . 024-6710144

Email: info@stekom.ac.id

Hak Cipta dilindungi Undang undang

Dilarang memperbanyak karya Tulis ini dalam bentuk dan dengan cara apapun
tanpa ijin tertulis dan penerbit.

KATA PENGANTAR

Mikrokontroler adalah suatu perangkat kecil yang mampu di gunakan untuk menangkap data dari sensor yang kemudian bisa di hubungkan dengan perangkat lain nya.

Pada perkembangan nya mikrokontroler juga dapat terkoneksi dengan perangkat lain nya dari jarak jauh dengan perantara teknologi Internet Of Things. Fungsi monitoring dan kontroling perangkat dengan jarak jauh akan dapat di lakukan dengan hal ini. Sebelum kita melangkah ke sana alangkah baik nya kita mengenal tentang dasar dari arduino

Buku ini kami susun dan untuk dipersembahkan bagi mahasiswa – mahasiswa Universitas STEKOM agar dengan mudah dan sesingkat memahami mengenai arduino dan penerapan nya dengan menggunakan berbagai macam sensor dengan di sertai contoh project DIY.

Harapan penyusun, buku ini bisa bermanfaat dan bisa digunakan oleh mahasiswa dan pembaca pada umumnya dan jangan segan – segan untuk mencoba dan mencoba karena dari situlah akan kita dapatkan suatu jawaban yang kita butuhkan.

Penulis

DAFTAR ISI

Cover	i
Halaman Judul	ii
Kata Pengantar	ii
BAB I Pengertian,Sejarah, dan Jenis-Jenis Arduino	1
A. Pengertian Arduino	1
B. Sejarah Arduino	2
C. Jenis-Jenis Arduino	3
D. Evaluasi	11
F. Referensi	12
BAB II Bahasa Pemrograman Arduino	13
A. Bahasa C	13
B. Struktur	15
C. Syntax	16
D. Variabel	16
E. Operator Perbandingan	17
F. Struktur Pengaturan	18
G. Pin Digital	18
H. Pin Analog	19
I. Tipe data	20
J. Evaluasi	21
K. Referensi	21
BAB III Cara pemakaian Arduino IDE	22
A. Arduino IDE	22
B. Board untuk Arduino	26
C. Contoh Project Led	28
D. Merubah Kecerahan Led.....	30
E. Multipel Led/Running Led	31
F. Evaluasi	36
G. Referensi	36
BAB IV Membaca Tegangan ADC	37
A. Membaca Tegangan ADC	37
B. Kaitan Nilai ADC dan Tegangan	38
C. Sketch Membaca ADC.....	39
D. Evaluasi	41
E. Referensi	42

BAB V	Sensor Temperatur LM35.....	43
	A. Pengertian LM35	43
	B. Cara Kerja LM 35	45
	C. Contoh Pemakaian LM 35	46
	D. Evaluasi.....	47
	E. Referensi	47
BAB VI	Sensor Cahaya LDR	48
	A. Pengertian LDR	48
	B. Prinsip Kerja LDR.....	50
	C. Contoh Penggunaan LDR.....	51
	D. Evaluasi	53
	E. Referensi	53
BAB VII	Sensor Pendeteksi Manusia PIR.....	54
	A. Pengertian PIR	54
	B. Spesifikasi	56
	C. Contoh Penggunaan PIR	57
	D. Evaluasi.....	60
	E. Referensi	60
BAB VIII	Sensor Ultrasonik HCSR 04.....	61
	A. Pengertian Sensor Ultrasonik	61
	B. Cara Kerja Sensor Ultrasonik	61
	C. Contoh Penggunaan Sensor Ultrasonik	63
	D. Evaluasi	70
	C. Referensi.....	71
BAB IX	Sensor Pengukuran Kelembaban Tanah	72
	A. Pengertian Soil Moisture.....	72
	B. Spesifikasi	73
	C. Cara Penggunaan Soil Moisture.....	75
	D. Evaluasi	77
	C. Referensi.....	77
BAB X	Motor Servo.....	78
	A. Pengertian Motor Servo	78
	B. Macam-Macam Motor Servo	79
	C. Spesifikasi Motor Servo	79
	D. Komponen Motor Servo	80
	A. Prinsip Kerja Motor Servo	81
	F. Cara Mengendalikan Motor Servo.....	82
	G. Contoh Penggunaan Motor Servo	84
	H. Evaluasi	86
	I. Referensi	86

BAB XI	Sensor Alkohol MQ-3	87
	A. Pengertian Sensor MQ-3	87
	B. Spesifikasi	88
	C. Cara Kerja	88
	D. Penggunaan MQ-3	89
	E. Evaluasi	91
	F. Referensi	91
BAB XII	Sensor MQ-7 pendeteksi Karbon	92
	A. PengertianMQ-7	92
	B. Struktur dan Konfigurasi MQ-7	93
	C. Contoh Penggunaan MQ-7	94
	D. Evaluasi.....	97
	E. Referensi.....	97
BAB XIII	Sensor Kelembaban Udara dan Temperatur DHT 11.....	98
	A. Pengertian Sensor Kelembaban	98
	B. Spesifikasi	99
	C. Contoh Pemakaian DHT 11 dan RTC.....	101
	D. Evaluasi	108
	E. Referensi.....	108
BAB XIV	Sensor Suara.....	109
	A. Pengertian sensor suara	109
	B. Contoh Penggunaan Sensor Suara KY-038	111
	C. Evaluasi	113
	D. Referensi	113
BAB XV	Project Helm Pintar Pendeteksi Gas.....	114
	A. Latar Belakang	114
	B. Alur Sistem.....	115
	C. Project Arduino DIY Helm Pintar.....	116
	D. Evaluasi	121
	A. Latar Referensi.....	123
BAB XVI	Project Penyiraman Tanaman Otomatis	123
	A. Latar Belakang	123
	B. Alur Kerja Sistem.....	124
	C. Project Penyiraman Otomatis Arduino DIY	125
	D. Evaluasi.....	128
	E. Referensi	123
	Daftar Pustaka.....	123

Bab 1. Pengertian, Sejarah, dan Jenis-jenis Arduino

Uraian :

Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai Pengetian, Sejarah serta jenis dan macam-macam arduino yang bisa di gunakan dan berserta fungsi nya.

Sasaran :

Mahasiswa mampu memahami dan mengerti mengenai dasar-dasar dari layout arduino dan segala macam pin yang di dalam nya.

A. Pengertian Arduino

Arduino adalah pengendali mikro single-board yang bersifat *open-source*, diturunkan dari *Wiring platform*¹, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor *Atmel AVR* dan softwrenya memiliki bahasa pemrograman sendiri.

Arduino juga merupakan *platform* hardware terbuka yang ditujukan kepada siapa saja yang ingin membuat purwarupa peralatan elektronik interaktif berdasarkan hardware dan software yang fleksibel dan mudah digunakan. Mikrokontroler diprogram menggunakan bahasa pemrograman arduino yang memiliki kemiripan *syntax* dengan bahasa pemrograman C. Karena sifatnya yang terbuka maka siapa saja dapat mengunduh skema hardware arduino dan membangunnya.

Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis, namun ada individu/perusahaan yang membuat *clone* arduino dengan menggunakan mikrokontroler lain dan tetap kompatibel dengan arduino pada level hardware. Untuk fleksibilitas, program dimasukkan melalui bootloader meskipun ada opsi untuk mem-bypass bootloader dan menggunakan downloader untuk memprogram mikrokontroler secara langsung melalui port ISP.

B. Sejarah

Semuanya berawal dari sebuah thesis yang dibuat oleh Hernando Barragan, di Institute Ivrea Italia pada tahun 2005, dikembangkan oleh Massimo Banzi dan David Cuartielles dan diberi nama *Arduin of Ivrea*. Lalu diganti nama menjadi Arduino yang dalam bahasa Italia berarti teman yang berani.

Tujuan awal dibuat Arduino adalah untuk membuat perangkat mudah dan murah, dari perangkat yang ada saat itu. Dan perangkat tersebut ditujukan untuk para siswa yang akan membuat perangkat desain dan interaksi.

Saat ini tim pengembangnya adalah Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis, dan Nicholas Zambetti. Mereka mengupayakan 4 hal dalam Arduino ini, yaitu:

1. Harga terjangkau
2. Dapat dijalankan diberbagai sistem operasi, Windows, Linux, Mac, dan sebagainya.
3. Sederhana, dengan bahasa pemrograman yang mudah bisa dipelajari orang awam, bukan untuk orang teknik saja.
4. Open Source, hardware maupun software.

Sifat Arduino yang Open Source, membuat Arduino berkembang sangat cepat. Sehingga banyak lahir perangkat-perangkat sejenis Arduino. Seperti DFRduino atau Freduino, sedangkan untuk lokal ada Cipaduino yang dibuat oleh SKIR70, lalu ada Murmerduino yang dibuat oleh Robot Unyil, ada lagi AViShaduino yang salah satu pembuatnya adalah Admin Kelas Robot.

Sampai saat ini pihak resmi, sudah membuat berbagai jenis-jenis Arduino. Mulai dari yang paling mudah dicari dan paling banyak digunakan, yaitu Arduino Uno. Hingga Arduino yang sudah menggunakan ARM Cortex, berbentuk

Mini PC. Hingga saat ini sudah ada ratusan ribu Arduino yang digunakan di dunia sejak tahun 2011. Arduino juga sudah dipakai oleh perusahaan-perusahaan besar, contohnya Google menggunakan Arduino untuk Accessory Development Kit, NASA memakai Arduino untuk prototipin, ada lagi Large Hadron Colider memakai Arduino dalam beberapa hal untuk pengumpulan data.

Banyak yang bertanya Arduino ini sebenarnya menggunakan bahasa pemrograman apa? Arduino sebenarnya menggunakan bahas C, yang sudah di sederhanakan. Sehingga orang awam pun bisa menjadi seniman digital, bisa mempelajari Arduino dengan mudahnya.

C. Jenis-Jenis Arduino

Dan seperti Microcontroller yang banyak jenisnya, Arduino lahir dan berkembang, kemudian muncul dengan berbagai jenis. Diantaranya adalah:

a) Arduino Uno

Jenis Arduino yang paling sering digunakan. Terutama untuk pemula atau media pembelajaran sangat disarankan menggunakan Arduino Uno. Selain banyaknya referensi yang membahas jenis arduino yang satu ini, juga karena chip mikrokontroller yang digunakan memakai jenis DIL / DIP (Dual In-Line Package). Sangat memudahkan pengguna mengganti chip mikrokontroller, jika terjadi kerusakan, dan juga kompatible dengan banyak Shield tambahan seperti, Ethernet, SD-CARD, GSM,dll. Versi yang terakhir adalah Arduino uno R3 (Revisi 3), menggunakan chip mikrokontroller Atmel AVR ATMEGA328, memiliki 14 pin I/O digital (6 diantaranya pin PWM), 6 pin input analog, . Komunikasi USB A to USB B (USB Printer) memudahkan komunikasi hardware dengan perangkat komputer / laptop.



Gambar1.Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

b) Arduino Leonardo

Arduino satu ini dibisa di bilang, kembaran Arduino uno, walaupun secara bentuk mirip. Perbedaan paling menonjol terdapat pada konektor USB dimana Arduino leonardo menggunakan konektor Mikro USB. Dan perbedaan lainya terletak pada chip mikrokontroller yang digunakan adalah ATMEGA32u4, memiliki 20 digital I/O (7 diantaranya pin PWM dan 12 Analog input), namun yang digunakan hanya sebagian yang disesuaikan dengan standar Arduino. Menurut saya Arduino Leonardo, kurang cocok digunakan untuk pemula atau media belajar, karena menggunakan chip mikrokontroller SMD (Surface-Mount Device). Jika terjadi kerusakan chip, akan sulit untuk menggantinya, karena membutuhkan keahlian khusus untuk melepas dan memasang kembali chip SMD



Gambar 2.Arduino Leonardo

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (Recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
Lenght	68.6 mm
Width	53.3 mm
Weight	20 g

c) Arduino due

Arduino Due adalah varian papan pengembang mikrokontroler Arduino yang menggunakan CPU **Atmel SAM3X8E ARM Cortex-M3**. Dengan demikian, Arduino Due adalah *Arduino Development Board* pertama yang didasarkan pada mikrokontroler ARM 32-bit. Arduino Due yang dirilis tahun 2012 dengan CPU 32-bit ARM Cortex-M3 sama sekali berbeda dari Arduino Duemilanove dengan MCU 8-bit ATmega168 yang dirilis tahun 2009. Miskonsepsi yang sering terjadi karena "Due" disalah-artikan sebagai singkatan dari *Duemilanove*. Due adalah *advanced board* baru dengan kemampuan jauh di atas

varian Arduino lainnya, Duemilanove adalah entry-level board lama yang merupakan "pendahulu" Arduino Uno.

Papan pengembang ini memiliki pin masukan/keluaran digital sebanyak 54 pin (12 di antaranya berkemampuan PWM), 12 pin masukan analog, 4 UART / hardware serial port, pencacah-waktu / clock berfrekuensi 84 MHz, koneksi dengan kemampuan USB OTG, 2 DAC (digital-to-analog converter), 2 TWI (Two Wire Interface, kompatibel dengan protokol I2C dari Phillips), soket jack catu daya standar (5,5/2,1mm), konektor SPI header, konektor JTAG header, tombol reset, dan sebuah tombol hapus (erase button).

Kelebihan utama Arduino Due dibanding varian Arduino lainnya terletak pada penggunaan CPU ARM Cortex-M3 yang memiliki fitur sbb

1. Core processor 32-bit yang memungkinkan operasi data sebanyak 4 byte sekaligus (tipe data DWORD) pada satu siklus waktu.
2. Jauh lebih cepat dengan frekuensi CPU clock 84 MHz RAM statis / SRAM sebesar 96 KiloByte (48 kali lipat lebih besar dibanding Arduino Uno, 12 kali lipat lebih besar dibanding Arduino Mega 2560)
3. Ruang untuk kode program / Flash Memory sebesar 512 KB (16 kali lipat lebih besar dibanding Arduino Uno, 2 kali lipat lebih besar dibanding Arduino Mega 2560).
4. Terdapat pengendali akses memori langsung (Direct Memory Access / DMA controller) yang dapat membebaskan CPU dari operasi memori yang intensif.
5. Terdapat dua kanal DAC (Digital-Analog-Converter) terpadu (Arduino lainnya memiliki ADC tapi tidak memiliki DAC yang merupakan komplemen fungsi dari ADC -- ADC mengubah sinyal analog menjadi

digital, DAC mengubah sinyal digital menjadi sinyal analog yang sesungguhnya / true analog. Bedakan dengan PWM yang men-simulasi-kan keluaran analog).

6. Resolusi ADC yang lebih presisi hingga 12-bit ($2^{12} = 4096$ jenjang, 0-4095) sebanyak 12 kanal
7. Semua pin I/O dapat difungsikan sebagai hardware external interrupt trigger.

Sumber catu daya dapat diambil dari port USB atau catu daya eksternal antara 6 ~ 16 Volt DC (direkomendasikan antara 7 ~ 12 VDC). Untuk membuat program, Anda dapat menggunakan Arduino IDE versi 1.5x (bukan versi 1.0 seperti pada Arduino lainnya) yang dapat diunduh secara gratis dari website resmi Arduino.

PENTING: Berbeda dengan papan pengembang Arduino lainnya, Arduino Due beroperasi pada tingkat tegangan 3,3V (sama seperti Arduino Pro Mini 3v3 / Raspberry Pi). Memasok dengan tegangan lebih tinggi (seperti 5V) ke pin I/O dapat merusak papan ini. Bila perlu menghubungkan dengan peralatan / modul yang bekerja dengan sinyal TTL 5 Volt, Anda dapat menggunakan 5V-3v3 level converter

Kekurangan Arduino Due dibanding varian Arduino lainnya yang menggunakan MCU ATmega adalah tidak adanya EEPROM terpadu. Apabila Anda membutuhkan EEPROM, Anda dapat menggunakan modul EEPROM tambahan sebesar 32 KB (32 kali lipat kapasitas EEPROM pada Arduino Uno) menggunakan AT24C256 EEPROM Module



Gambar 3.Arduino Due

d) Arduino Mega 2560

Arduino mega 2560 adalah papan mikrokontroler ATmega2560 berdasarkan (datasheet) memiliki 54 digital pin input / output (dimana 15 dapat digunakan sebagai output PWM), 16 analog input, 4 UART (hardware port serial), osilator kristal 16 MHz, koneksi USB, jack listrik, header ICSP, dan tombol reset. Ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya menghubungkannya ke komputer dengan kabel USB atau power dengan adaptor AC-DC atau baterai. Arduino Mega kompatibel dengan sebagian besar shield,dirancang untuk Arduino Duemilanove atau Diecimila. Arduino Mega2560 berbeda dari semua board sebelumnya ,tidak menggunakan chip driver FTDI USB-to-serial. Sebaliknya, fitur ATmega16U2 (ATmega8U2 dalam revisi 1 dan revisi 2 papan) diprogram sebagai konverter USB-to-serial. Revisi 2 dewan Mega2560 memiliki resistor menarik garis 8U2 HWB ke tanah, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU.

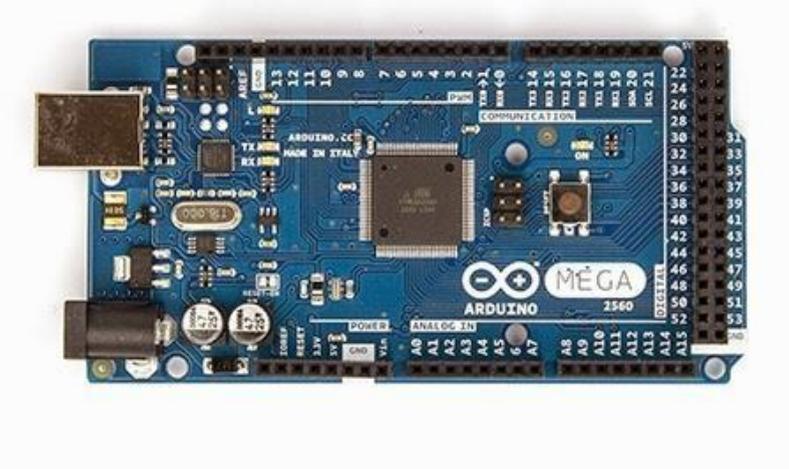
Revisi 3 dari dewan memiliki fitur-fitur baru berikut:

- 1,0 pinout: menambahkan SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, yang IOREF yang memungkinkan perisai untuk beradaptasi dengan tegangan yang tersedia dari

papan. Di masa depan, perisai akan kompatibel baik dengan dewan yang menggunakan AVR yang beroperasi dengan 5V dan dengan Arduino Due yang beroperasi dengan 3.3V. Yang kedua adalah pin tidak terhubung, yang disediakan untuk tujuan masa depan.

-Stronger RESET sirkuit.

-Atmega 16U2 menggantikan 8U2.



Gambar 4. Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide

	PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

D. Evaluasi

1. Sebutkan 3 jenis arduino yang anda ketahui ?
2. Terdapat pin apa saja dalam arduino Uno ?Jelaskan kegunaannya !
3. Sebutkan arduino yang mempunyai pin lebih dari 32 !

E. Referensi

1. Abdul Kadir (2017). *Pemrograman Arduino dan Prossesing*, PT.Elex Media Komputindo,Jakarta
2. Harlod Timmis (2011). *Practical Arduino Engineering*. Apress 1st edition.

3. Yuwono Martadinata., (2016). *Arduino itu Pintar ,PT Elex Media Komputindo, Jakarta*
4. Hari Santoso (2015). *Panduan Praktis Arduino untuk Pemula. Elang Sakti. Malang*

BAB 2 Bahasa Pemrograman Arduino

Uraian :

Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai dasar-dasar pemrograman arduino dan macam-macam tipe data yang di gunakan dalam arduino

Sasaran :

Mahasiswa mampu memahami konsep bahasa C yang di gunakan untuk memprogram arduino dan struktur data nya.

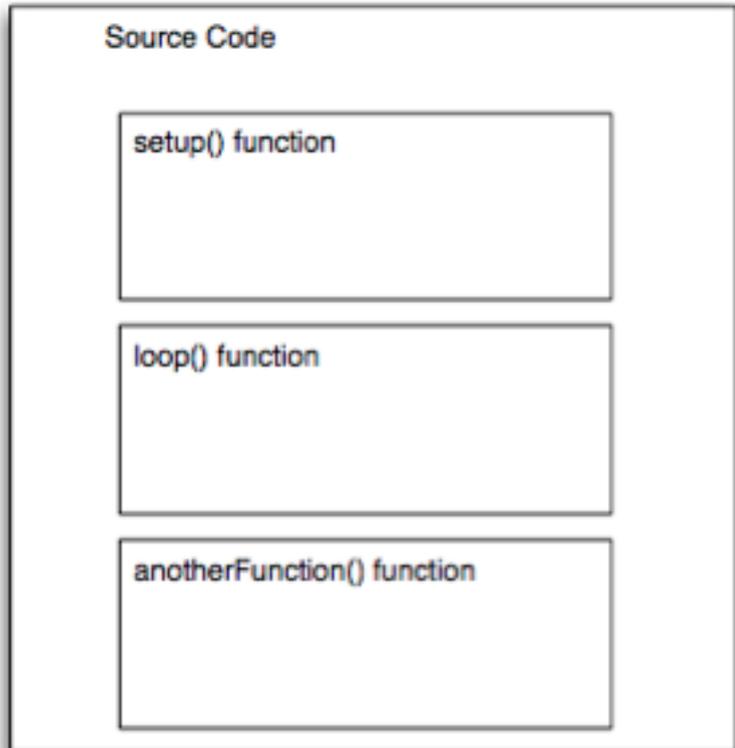
A. Bahasa C

Arduino menggunakan pemrograman dengan bahasa C. Berikut ini adalah sedikit penjelasan yang ditujukan kepada anda yang hanya mempunyai sedikit pengalaman pemrograman dan membutuhkan penjelasan singkat mengenai karakter bahasa C dan software Arduino. Untuk penjelasan yang lebih mendalam, web Arduino.cc adalah sumber yang lengkap.

Semua bahasa perograman terdiri dari

1. ekspresi,
 2. statemen,
 3. blok statemen
 4. blok fungsi
- **Ekspresi** adalah kombinasi operand dan operator contoh $2+3$, $X > Y$ dst... 2,3,X dan Y adalah operand sedangkan + dan > adalah operator .
 - **Statemen** adalah instruksi lengkap dalam bahasa C diakhiri dgn tanda ; (titik koma) contoh : $A= 2+3;$

- **Statemen dan ekspresi** C arduino indentik dengan ANSI-C , Struktur Program C Arduino minimal terdiri dari dua fungsi yaitu `setup()` dan `loop()`.



Gambar 7. Susunan dalam bahasa yang di gunakan dalam arduino

- Fungsi `setup()` dijalankan pertama kali setiap board arduino dihidupkan sedangkan fungsi `loop()` dijalankan terus menerus selama board arduino hidup. Pada program standar C ANSI fungsi yg pertama dijalankan adalah fungsi `main()` pada C arduino adalah fungsi `setup()`.
- berikut ini struktur minimal C Arduino :

```
//Setup digunakan untuk inisialisasi variable, mode pin dll
Void setup()
```

```

{
Statement
}
Void loop()
{
Statement
}

```

Structure	Variables	Functions
<ul style="list-style-type: none"> + <code>setup()</code> + <code>loop()</code> <p>Control Structures</p> <ul style="list-style-type: none"> + <code>if</code> + <code>if...else</code> + <code>for</code> + <code>switch case</code> + <code>while</code> + <code>do... while</code> + <code>break</code> + <code>continue</code> + <code>return</code> + <code>goto</code> 	<p>Constants</p> <ul style="list-style-type: none"> + <code>HIGH LOW</code> + <code>INPUT OUTPUT</code> + <code>true false</code> + integer constants + floating point constants <p>Data Types</p> <ul style="list-style-type: none"> + <code>void</code> + <code>boolean</code> + <code>char</code> + <code>unsigned char</code> + <code>byte</code> + <code>int</code> 	<p>Digital I/O</p> <ul style="list-style-type: none"> + <code>pinMode()</code> + <code>digitalWrite()</code> + <code>digitalRead()</code> <p>Analog I/O</p> <ul style="list-style-type: none"> + <code>analogReference()</code> + <code>analogRead()</code> + <code>analogWrite() - PWM</code> <p>Advanced I/O</p> <ul style="list-style-type: none"> + <code>tone()</code> + <code>noTone()</code> + <code>shiftOut()</code> + <code>shiftIn()</code>

Gambar 8. Beberapa perintah yang di pake di arduino

B. Struktur

Setiap program Arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada.

```
void setup() { }
```

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.

```
void loop() { }
```

Fungsi ini akan dijalankan setelah `setup` (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.

C. Syntax

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan.

//(komentar satu baris)

- a. Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.

/* */(komentar banyak baris)

- b. Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

{ }(kurung kurawal)

- c. Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

;(titik koma)

- d. Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

D. Variable

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memindahkannya.int

(integer)

Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka desimal dan menyimpan nilai dari -32,768

long (long)

Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari -2,147,483,648 dan 2,147,483,647.

boolean (boolean)

Variabel sederhana yang digunakan untuk menyimpan nilai *TRUE* (benar) atau *FALSE* (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.

float (float)

Digunakan untuk angka desimal (floating point). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang dari -3.4028235E+38 dan 3.4028235E+38.
char (character)
Menyimpan 1 karakter menggunakan kode ASCII (misalnya 'A' = 65). Hanya memakai 1 byte (8 bit) dari RAM.

E. Operator Perbandingan

Digunakan untuk membandingkan nilai logika.

==

Sama dengan (misalnya: 12 == 10 adalah FALSE (salah) atau 12 == 12 adalah TRUE (benar))

!=

Tidak sama dengan (misalnya: 12 != 10 adalah TRUE (benar) atau 12 != 12 adalah FALSE (salah))

<

Lebih kecil dari (misalnya: 12 < 10 adalah FALSE (salah) atau 12 < 12 adalah FALSE (salah) atau 12 < 14 adalah TRUE (benar))

>

Lebih besar dari (misalnya: $12 > 10$ adalah TRUE (benar) atau $12 > 12$ adalah FALSE (salah) atau $12 > 14$ adalah FALSE (salah))

F. Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan (banyak lagi yang lain dan bisa dicari di internet).

1. if..else, dengan format seperti berikut ini:

```
if (kondisi) { }  
else if (kondisi) { }  
else { }
```

Dengan struktur seperti diatas program akan menjalankan kode yang ada di dalam kurung kurawal jika kondisinya TRUE, dan jika tidak (FALSE) maka akan diperiksa apakah kondisi pada *else if* dan jika kondisinya FALSE maka kode pada *else* yang akan dijalankan.

2. for, dengan format seperti berikut ini:

```
for (int i = 0; i < #pengulangan; i++) { }
```

Digunakan bila anda ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan penghitungan ke atas dengan $i++$ atau ke bawah dengan $i--$.

G. Pin Digital

1. pinMode(pin, mode)

Digunakan untuk menetapkan mode dari suatu pin, *pin* adalah nomor pin yang akan digunakan dari

0-19 (pin analog 0-5 adalah 14-19). Mode yang bisa digunakan adalah *INPUT* atau *OUTPUT*.

2. digitalWrite(pin, value)

Ketika sebuah pin ditetapkan sebagai *OUTPUT*, pin tersebut dapat dijadikan *HIGH* (ditarik menjadi 5 volts) atau *LOW* (diturunkan menjadi ground).

3. digitalRead(pin)

Ketika sebuah pin ditetapkan sebagai *INPUT* maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah *HIGH* (ditarik menjadi 5 volts) atau *LOW* (diturunkan menjadi ground).

H. Pin Analog

Arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam alam analog (menggunakan trik). Berikut ini cara untuk menghadapi hal yang bukan digital.

1. analogWrite(pin, value)

Beberapa pin pada Arduino mendukung PWM (pulse width modulation) yaitu pin 3, 5, 6, 9, 10, 11. Ini dapat merubah pin hidup (*on*) atau mati (*off*) dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluaran analog. *Value* (nilai) pada format kode tersebut adalah angka antara 0 (0% duty cycle ~ 0V) dan 255 (100% duty cycle ~ 5V).

2. analogRead(pin)

Ketika pin analog ditetapkan sebagai *INPUT* anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 volts) dan 1024 (untuk 5 volts).

I. Type Data

Arduino C mendukung sebagian besar tipe data ANSI C dengan beberapa pengecualian. Variabel adalah sebuah atau beberapa lokasi memori yang diberi nama. Ketika anda mendefinisikan variabel Anda juga harus memberitahu compiler jenis data variable tsb.

Tipe data dari variabel penting karena untuk menentukan berapa banyak byte memori yg disediakan untuk variabel itu, dan jenis data yang dapat disimpan dalam variable

Type	Size in Bits	Range
char	8	-127 to 127
unsigned char	8	0 to 255
signed char	8	-127 to 127
int	16	-32,767 to 32,767
unsigned int	16	0 to 65,535
signed int	16	-32,767 to 32,767
short int	16	-32,767 to 32,767
unsigned short int	16	0 to 65,535
signed short int	16	-32,767 to 32,767
long int	32	-2,147,483,647 to 2,147,483,647
long long int	64	$-(2^{63} - 1)$ to $2^{63} - 1$ (Added by C99)
signed long int	32	-2,147,483,647 to 2,147,483,647
unsigned long int	32	0 to 4,294,967,295
unsigned long long int	64	$2^{64} - 1$ (Added by C99)
float	32	1E-37 to 1E+37 with six digits of precision
double	64	1E-37 to 1E+37 with ten digits of precision
long double	80	1E-37 to 1E+37 with ten digits of precision

Gambar 8. Type Data yang dipakai dalam arduino

J. Evaluasi

1. Apa yang di maksud dengan Analog Read ?
2. Apa yang di maksud tipe data Boolean ?
3. Apa fungsi void loop ?

K. Referensi

1. Abdul Kadir (2017). *Pemrograman Arduino dan Prossesing*, PT.Elex Media Komputindo,Jakarta
2. Harlod Timmis (2011). *Practical Arduino Engineering*. Apress 1st edition.
3. Yuwono Martadinata., (2016). *Arduno itu Pintar* ,PT Elex Media Komputindo, Jakarta
4. Hari Santoso (2015). *Panduan Praktis Arduino untuk Pemula*. Elang Sakti. Malang

Bab 3.Cara Pemakaian Arduino IDE

Uraian :

Mata kuliah ini memberikan pengetahuan kepada mahasiswa bagaimana menggunakan software editor Arduino IDE dan mengupload nya ke dalam board Arduio

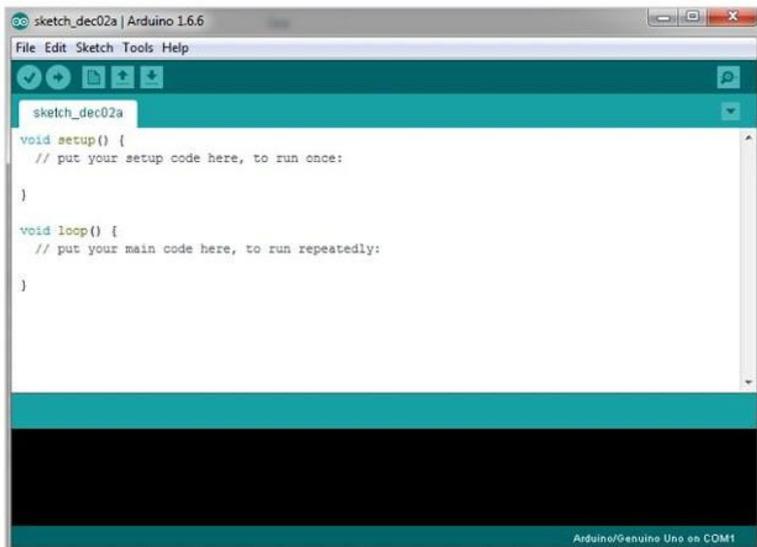
Sasaran :

Mahasiswa mampu memahami menggunakan Arduino IDE dan Mengupload ke dalam Arduino sehingga bisa mengkontrol mikrokontroler dan sensor

A. Arduino IDE

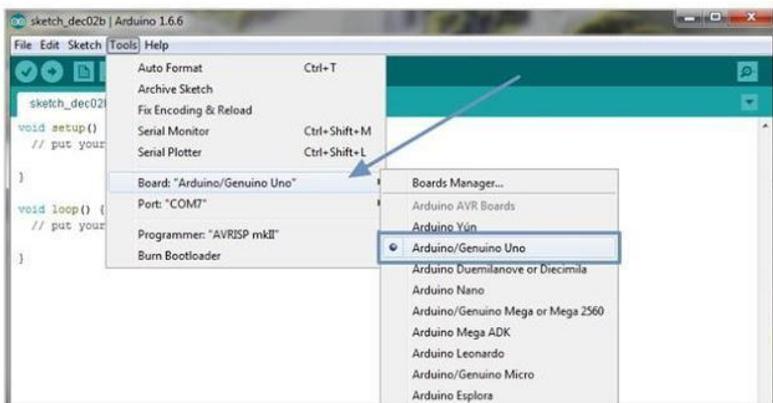
Sebelum sampai pada tahap pembuatan dan upload *sketch* (program) ke board arduino, dibutuhkan sedikit pengaturan pada IDE terlebih dahulu. Pengaturan ini adalah terkait **jenis board** yang digunakan (dalam kasus ini adalah arduino uno) dan pengaturan **port**. Untuk mulai melakukan pengaturan ini, buka software IDE arduino .

Untuk membuka IDE arduino, double-klik pada shortcut Arduino seperti yang ditunjukkan pada gambar diatas. IDE arduino akan terbuka dengan antarmuka seperti dibawah ini:



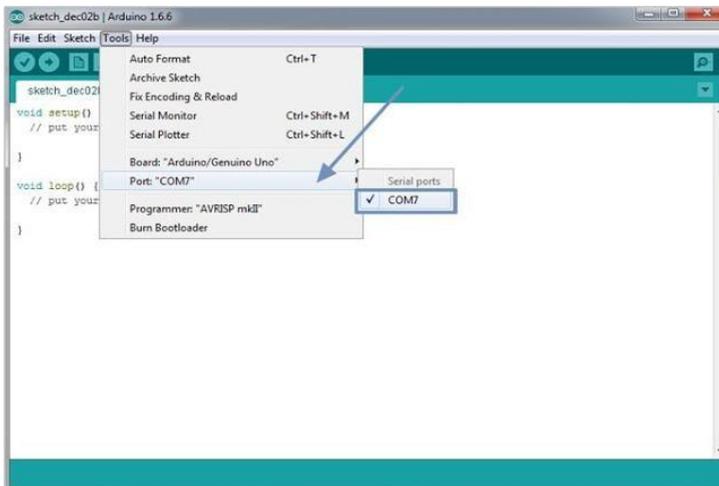
Gambar 9. Tampilan awal dari IDE Arduino

Terdapat 5 (lima) menu utama yang terdapat pada bagian atas IDE yaitu file, edit, sketch, tools dan help. Untuk men-setting board yang digunakan, Klik pada menu “**Tools**”, pada bagian “**Board**”, pilih sesuai dengan board arduino yang digunakan (dalam kasus ini adalah Arduino Uno).



Gambar 10 . Ketika Tools di tekan untuk memilih board

Selanjutnya adalah setting port, pada bagian “**Port**”, pilih port yang tersedia (dalam kasus ini adalah COM7). Untuk beberapa keadaan, nama port yang ditampilkan bisa berbeda. Jika port belum terdeteksi, koneksikan arduino terlebih dahulu lalu buka sketch



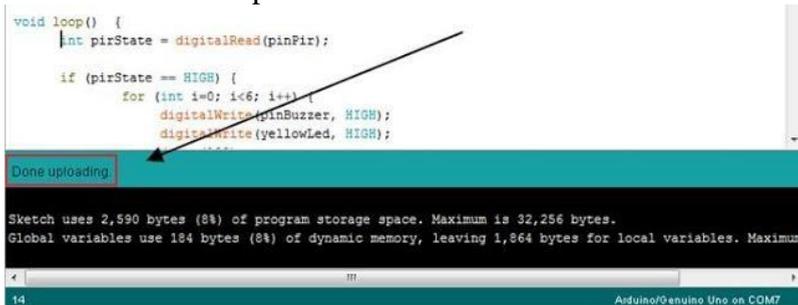
Gambar 11. Pemilihan jenis port yang di pakai.

Ketika pemilihan Port yang akan di gunakan Upload sketch adalah tahap dimana program ditanamkan pada board arduino. Pada tutorial ini, diasumsikan bahwa anda telah selesai menulis program yang akan ditanamkan pada board arduino. Untuk mulai meng-upload program, hubungkan board arduino ke PC melalui USB, klik pada icon **Panah Kanan** seperti yang ditunjukkan pada gambar dibawah ini:



Gambar 12. Tombol untuk upload data ke arduino

Proses upload akan berjalan selama beberapa detik. Jika tidak ada kesalahan pada kode program yang dibuat, akan ada notifikasi **“Done uploading”** yang menandakan bahwa sketch berhasil di-upload ke board arduino.

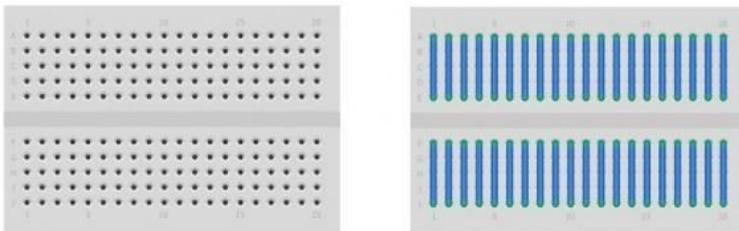


Gambar 13. Proses ketika sudah berhasil upload

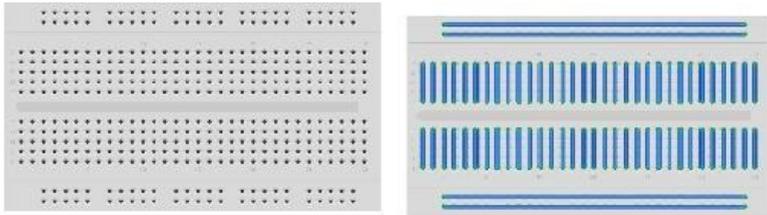
B. Breadboard untuk Arduino

Breadboard adalah board yang digunakan untuk membuat rangkaian elektronik sementara dengan tujuan uji coba atau prototipe tanpa harus menyolder. Dengan memanfaatkan breadboard, komponen-komponen elektronik yang dipakai tidak akan rusak dan dapat digunakan kembali untuk membuat rangkaian yang lain. Breadboard umumnya terbuat dari plastik dengan banyak lubang-lubang di atasnya. Lubang-lubang pada breadboard diatur sedemikian rupa membentuk pola sesuai dengan pola jaringan koneksi di dalamnya. Breadboard yang tersedia di pasaran umumnya terbagi atas 3 ukuran: **mini breadboard, medium breadboard atau large breadboard**. Mini breadboard memiliki 170 titik koneksi (bisa juga lebih). Kemudian medium breadboard memiliki 400 titik koneksi. Dan large breadboard memiliki 830 titik koneksi.

Perhatikan gambar di bawah, sebuah mini breadboard dengan 200 titik koneksi. Pada bagian kanan dapat dilihat pola layout koneksi yang digambar dengan garis berwarna biru



Gambar 14. Gambar breadboard mini



Gambar 15. Breadboard Medium

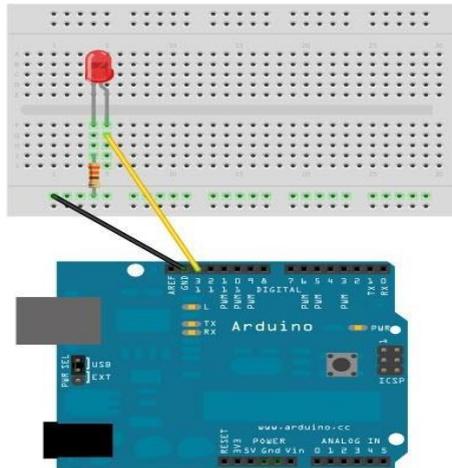
Gambar diatas adalah medium breadboard dengan 400 titik koneksi. Medium breadboard ini biasa juga disebut half (setengah) breadboard. Karena ukurannya kurang lebih setengah dari ukuran large / full breadboard dengan 830 titik koneksi.

Pada breadboard tersebut dapat dilihat penulisan huruf A, B, C, D, E, F, G, H, I dan J. Kemudian ada angka 1, 5, 10, 15 dan 20. Huruf dan angka ini membentuk semacam koordinat. A1, B1, C1, D1 dan E1 saling berhubungan sesuai pola koneksinya (lihat kembali garis berwarna biru).

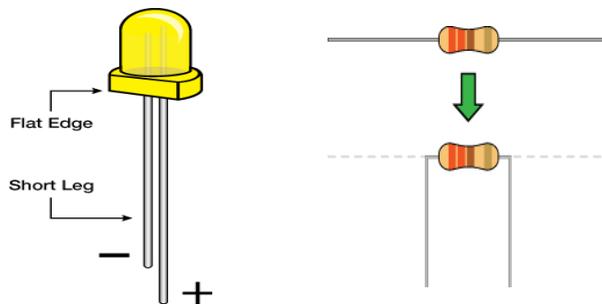
Begitu juga A2 → E2, A3 → E3, F1 → J1, F2 → J2 dan seterusnya. Dengan memahami pola koneksi ini kita sudah bisa memakai breadboard untuk keperluan prototipe rangkaian sehingga dapat menempatkan komponen elektronik secara tepat sesuai gambar rangkaian yang dimaksud.

C. Contoh Project LED

Setelah mengetahui dasar-dasar tentang breadboard, kita coba menggunakannya dengan Arduino. Rangkaian dasar yang biasa dipakai untuk belajar Arduino adalah **Blinking LED**



Gambar 16. Rangkaian Blinking LED



Gambar 17. Led dan Resistor

Script

Di bawah ini adalah kode pemrograman yang digunakan untuk blinking LED. Tidak perlu mengetiknya karena sudah tersedia pada software Arduino. Klik menu **File > Examples > 1.Basic > Blink.**

```
/*  
  
    Blink Turns on an LED on for one second, then off  
    for one second, repeatedly.  
  
    This example code is in the public domain.  
  
    */  
  
void setup() {  
    // initialize the digital pin as an output.  
  
    // Pin 13 has an LED connected on most Arduino  
    boards:  
  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH); // set the LED on  
    delay(1000);           // wait for a second  
    digitalWrite(13, LOW); // set the LED off  
    delay(1000);           // wait for a second  
  
}
```

- **Membuat Perubahan**
- Bila tidak terdapat kesalahan, maka rangkaian anda akan berupa LED yang berkelap-kelip dengan durasi hidup 1 detik dan mati 1 detik yang dilakukan berulang-ulang. Sudah puas? Hehehe rasanya pasti belum. Bagaimana kalau kita melakukan beberapa

perubahan pada program agar kita dapat lebih memahami cara kerjanya.

- Beberapa perubahan yang bisa kita lakukan, antara lain:
- **Mengganti pin**
- Pada program diatas, LED kita terhubung pada pin 13, tetapi kita bisa menggunakan pin Arduino yang lain. Untuk menggantinya, ambil kabel jumper yang terpasang pada pin 13 dan pindahkan pada pin lain sesuai yang anda inginkan (dari 0-13) (anda juga bisa menggunakan pin analog 0-5, pin 0 analog adalah pin 14, dst).
- Kemudian rubah baris kode program, ganti angka 13 dengan pin baru yang sudah dipilih.
- Lalu upload sketch-nya (ctrl+U)Merubah durasi kelap-kelip Kalau anda kurang puas dengan kelap-kelip berdurasi 1 detik, berikut caranya:

Rubah baris kode program:

```
digitalWrite(13, HIGH);  
delay(durasi hidup);           //(detik * 1000)
```

```
digitalWrite(13, LOW);  
delay(durasi mati);           //(detik * 1000)
```

Ganti durasi hidup/mati dengan durasi yang anda inginkan. Misalkan untuk 5 detik, tuliskan angka 5000 (5 * 1000)

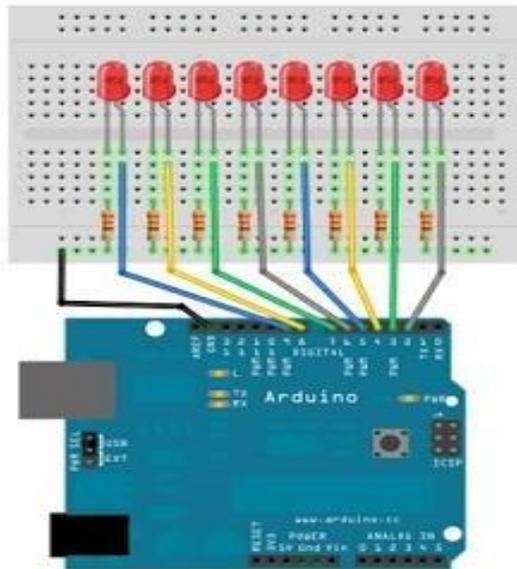
D. Merubah cerahnya LED (brightness)

- Selain pengaturan digital yang hidup atau mati, Arduino dapat juga melakukan pengaturan beberapa pin digital seperti layaknya analog yang akan kita gunakan untuk mengatur cerahnya LED. Untuk mencobanya, lakukan langkah berikut ini:
- Rubah pin LED menjadi pin 9 (jangan lupa rubah juga kabel jumpernya). Rubah baris kode program, ganti 13 menjadi angka 9.

- Rubah kode dalam { } pada bagian **loop()** dengan kode berikut:
- ```
void loop() {
 analogWrite(9,angkabaru);
}
```
- angka baru = angka antara 0 – 255. 0 = mati, 255 = hidup, angka diantaranya = perbedaan tingkat kecerahan

#### **E. Multiple LEDs / running led**

- Sejalan dengan pengaturan LED kita akan mencoba untuk membuat kode pemrograman lebih sederhana dengan beberapa metode.
- for () loops – digunakan ketika kita ingin menjalankan sejumlah baris kode berulang-ulang.
- arrays[] – digunakan untuk mengatur penggunaan variabel dengan lebih mudah (array = grup dari variabel).
- Rangkaian  
Komponen yang diperlukan:  
8 buah LED  
8 buah resistor 330 ohm  
Kabel jumper



Gambar 17. Skema Running Led

**Menghidupkan LED satupersatu dari PIN 2 ke 9 dan mematikan dari PIN 9 ke 2**

```
//LED Pin Variables
int ledPins[] = {2,3,4,5,6,7,8,9}; //An array to hold
the pin each LED is connected to //i.e. LED #0 is
connected to pin 2, LED #1, 3 and so on //to address
an array use ledPins[0] this would equal 2 and
ledPins[7] would equal 9
```

```
/*
 * setup() - this function runs once when you turn
your Arduino on
 * We the three control pins to outputs
 */
void setup()
{
```

```
//Set each pin connected to an LED to output mode
(pulling high (on) or low (off))
for(int i = 0; i < 8; i++){ //this is a loop and will
repeat eight times
 pinMode(ledPins[i],OUTPUT); //we use this to
set each LED pin to output
} //the code this replaces is below
```

```
/* (commented code will not run)
* these are the lines replaced by the for loop above
they do exactly the
```

```
* same thing the one above just uses less typing
```

```
pinMode(ledPins[0],OUTPUT);
```

```
pinMode(ledPins[1],OUTPUT);
```

```
pinMode(ledPins[2],OUTPUT);
```

```
pinMode(ledPins[3],OUTPUT);
```

```
pinMode(ledPins[4],OUTPUT);
```

```
pinMode(ledPins[5],OUTPUT);
```

```
pinMode(ledPins[6],OUTPUT);
```

```
pinMode(ledPins[7],OUTPUT);
```

```
(end of commented code)*/
```

```
}
```

```
* loop() - this function will start after setup finishes
and then repeat
```

```
* we call a function called oneAfterAnother(). if you
would like a different behaviour
```

```
* uncomment (delete the two slashes) one of the
other lines
```

```
*/
```

```
void loop() // run over and over again
```

```
{
```

```
oneAfterAnotherNoLoop(); //this will turn on each
LED one by one then turn each off
```

```
}
```

```

/*
 * oneAfterAnotherNoLoop() - Will light one LED
then delay for delayTime then light
 * the next LED until all LEDs are on it will then turn
them off one after another
 *
 * this does it without using a loop which makes for a
lot of typing.
 * oneOnAtATimeLoop() does exactly the same thing
with less typing
 */
void oneAfterAnotherNoLoop(){
 int delayTime = 100; //the time (in milliseconds) to
pause between LEDs
 //make smaller for quicker switching
and larger for slower
 digitalWrite(ledPins[0], HIGH); //Turns on LED #0
 (connected to pin 2)
 delay(delayTime); //waits delayTime
 milliseconds
 digitalWrite(ledPins[1], HIGH); //Turns on LED #1
 (connected to pin 3)
 delay(delayTime); //waits delayTime
 milliseconds
 digitalWrite(ledPins[2], HIGH); //Turns on LED #2
 (connected to pin 4)
 delay(delayTime); //waits delayTime
 milliseconds
 digitalWrite(ledPins[3], HIGH); //Turns on LED #3
 (connected to pin 5)
 delay(delayTime); //waits delayTime
 milliseconds
 digitalWrite(ledPins[4], HIGH); //Turns on LED #4
 (connected to pin 6)

```

```

delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[5], HIGH); //Turns on LED #5
(connected to pin 7)
delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[6], HIGH); //Turns on LED #6
(connected to pin 8)
delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[7], HIGH); //Turns on LED #7
(connected to pin 9)
delay(delayTime); //waits delayTime
milliseconds

//Turns Each LED Off
digitalWrite(ledPins[7], LOW); //Turns off LED #7
(connected to pin 9)
delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[6], LOW); //Turns off LED #6
(connected to pin 8)
delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[5], LOW); //Turns off LED #5
(connected to pin 7)
delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[4], LOW); //Turns off LED #4
(connected to pin 6)
delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[3], LOW); //Turns off LED #3
(connected to pin 5)
delay(delayTime); //waits delayTime
milliseconds

```

```

digitalWrite(ledPins[2], LOW); //Turns off LED #2
(connected to pin 4)
delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[1], LOW); //Turns off LED #1
(connected to pin 3)
delay(delayTime); //waits delayTime
milliseconds
digitalWrite(ledPins[0], LOW); //Turns off LED #0
(connected to pin 2)
delay(delayTime); //waits delayTime
milliseconds
}

```

## **F. Evaluasi**

1. Bagaimana cara Membuat led berjalan berhenti sejenak ?
2. Apa yang harus di periksa ketika proses upload ke board gagal ?

## **G. Referensi**

1. Abdul Kadir (2017). *Pemrograman Arduino dan Prossesing, PT.Elex Media Komputindo, Jakarta*
2. Harlod Timmis (2011). *Practical Arduino Engineering. Apress 1<sup>st</sup> edition.*
3. Yuwono Martadinata., (2016). *Arduno itu Pintar ,PT Elex Media Komputindo, Jakarta*
4. Hari Santoso (2015). *Panduan Praktis Arduino untuk Pemula. Elang Sakti. Malang*

## **Bab 4. Membaca Tegangan ADC**

### ***Uraian :***

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai pembacaan tegangan ADC yang terdapat pada Arduino*

### ***Sasaran :***

*Mahasiswa mampu memahami dan menghitung nilai tegangan yang terdapat pada arduino untuk di gunakan menghitung nilai yang di peroleh dari sensor*

### **A. MembacaTegangan ADC**

Bagaimana cara menghitung nilai pembacaan ADC yang berasal dari suatu sensor analog. Microcontroller memiliki kemampuan untuk menerima, mengolah, dan memberikan output signal digital. Sebagai contoh bila microcontroller dengan catu daya 5 volt, microcontroller tersebut dapat mengenali tegangan sebesar 0 volt sebagai signal digital low, atau tegangan 5 volt sebagai signal digital high. Sedangkan untuk mengenali nilai tegangan antara 0 volt hingga 5 volt, diperlukan feature khusus, yakni Analog-to-Digital Converter. Analog-to-Digital Converter atau biasa disebut ADC, memungkinkan microcontroller untuk mengenali suatu nilai analog melalui suatu pendekatan digital. Saat ini banyak microcontroller, termasuk Arduino telah dilengkapi dengan feature ADC yang terintegrasi di dalamnya.

Tapi tidak semua pin Arduino dapat digunakan untuk mengolah signal analog. Pada board Arduino Uno, terdapat enam pin analog, yakni mulai dari A0 hingga A5. Huruf A pada awal nama pin Arduino menandakan pin tersebut dapat digunakan untuk mengolah signal analog. Seberapa tepat nilai signal analog yang dipetakan secara digital, ditentukan

oleh seberapa besar resolusi ADC. Semakin besar resolusi ADC, maka semakin mendekati nilai analog dari signal tersebut. Untuk resolusi ADC pada board Arduino Uno ialah 10 bit, yang berarti mampu memetakan hingga 1024 discrete analog level. Beberapa jenis microcontroller lain memiliki resolusi 8 bit, 256 discrete analog level, bahkan ada yang memiliki resolusi 16 bit, 65536 discrete analog level.

## **B. Kaitan Nilai ADC dengan Tegangan**

Nilai ADC menunjukkan ratio perbandingan dengan tegangan yang terbaca. Berikut persamaannya ialah nilai ADC terukur ialah nilai ADC maximum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber.

Nilai ADC tergantung dengan tegangan yang menjadi catu daya sistem microcontroller. Untuk board Arduino biasa menggunakan sumber tegangan 5 volt. Berikut adalah cara mencari nilai ADC, dengan menerapkan persamaan yang ada dan tegangan terbaca sebesar 2,12 volt pada board Arduino Uno.

Board Arduino Uno memiliki resolusi 10 bit, dengan nilai terbesar 1023. Tegangan sumber 5 volt dan tegangan terbaca ialah 2,12 volt. Nilai ADC terukur ialah nilai ADC maximum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber. Sehingga diperoleh nilai ADC sebesar 434

Dengan demikian diperoleh nilai ADC sebesar 434 dari tegangan terukur 2,12 volt. Untuk setiap unit ADC tersebut memiliki perbandingan tegangan sebesar 4,9 mV.

Proses konversi dari nilai analog menjadi digital ini disebut proses ADC (Analog to Digital Conversion). Bagaimana jika tegangan 5 volt dikonversi menjadi data digital 10 bit? Mari kita hitung

$$\frac{5 \text{ volt}}{1023} = 0,004887585$$

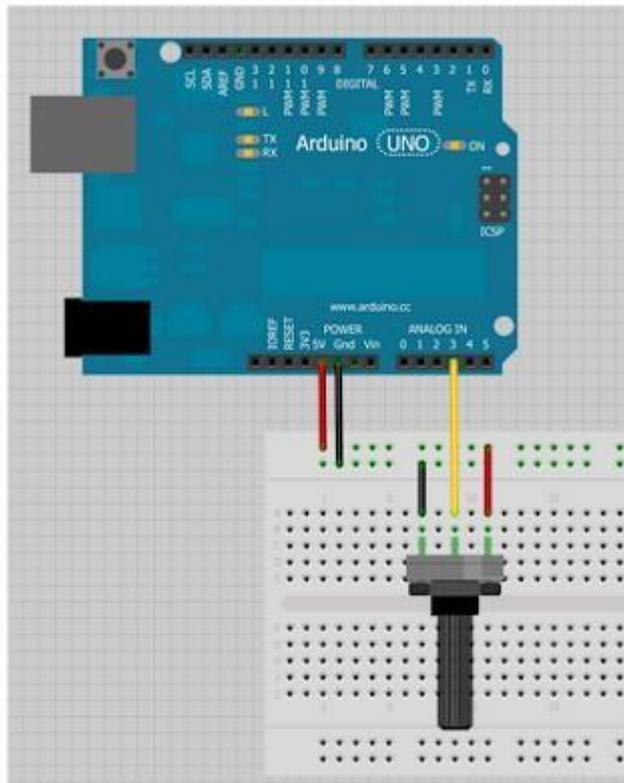
Artinya setiap 1 angka desimal mewakili tegangan sebesar 0,004887585 volt. Berapa besar tegangan yang diwakili angka 512?

$$512 \cdot \frac{5 \text{ volt}}{1023} = 2,50244 \text{ volt} \approx 2,5 \text{ volt}$$

Selain dipengaruhi oleh besarnya nilai resolusi ADC, tepat tidaknya pengukuran nilai ADC juga dipengaruhi oleh clock speed ADC tersebut. Untuk board Arduino Uno sendiri clock speed ADC maximum yang disarankan ialah 200 kHz. Nilai clock speed 200 kHz tersebut berdasarkan spesifikasi internal DAC (Digital to Analog Converter) pada rangkaian pengubahnya. Meski demikian, penggunaan clock speed pada 1 MHz mesalnya, tidak mengurangi kualitas resolusi ADC tersebut.

### **C. Sketch Membaca ADC**

Tutorial project DIY (Do It Yourself) sederhana untuk belajar membaca nilai ADC dapat dilakukan dengan menggunakan potentiometer. Potentiometer bekerja dengan prinsip pembagi tegangan, dimana terdapat kaki untuk sumber tegangan dihubungkan ke Vcc 5 volt Arduino, ground dihubungkan ke ground Arduino, dan data tegangan dihubungkan ke pin analog A3 Arduino.



**Gambar 17. Skematik arduino dan potensio**

/\*

Program membaca nilai  
ADC dengan serial

\*/

```
int analogPin =
A3; int lang =
0;
```

```
void setup()
```

```

{
 Serial.begin(9600);
}

void loop()
{
 lang =
 analogRead(analogPin);
 delay(100);
 Serial.println(lang);
}

```

Berikut ini adalah penjelasan untuk source code (sketch) program tersebut.

- Untuk pin analog yang digunakan ialah A3, yang terhubung dengan data analog dari potentiometer
- Untuk variable lang dengan (jenis data integer) digunakan untuk menyimpan nilai pembacaan ADC sebelum ditampilkan, dan diberikan nilai awal 0
- Proses inisialisasi setup dengan menentukan baud rate yang digunakan untuk jalur serial, yakni 9600
- Untuk program utama berisi function analogRead() yang digunakan untuk membaca nilai ADC pada analogPin, yakni pin Arduino A3
- Pemberian delay() ditujukan agar pembacaan nilai ADC stabil
- Baris program selanjutnya untuk menampilkan data nilai ADC pada serial monitor

#### **D. Evaluasi**

1. Apa yang di maksud dengan ADC ?

2. Menggunakan Pin brapa untuk membaca tegangan ADC ?
3. Apa Kaitan nya ADC dan Tegangan ?

#### **E. Referensi**

1. Abdul Kadir (2017). *Pemrograman Arduino dan Prossesing*, PT.Elex Media Komputindo, Jakarta
2. Harlod Timmis (2011). *Practical Arduino Engineering*. Apress 1<sup>st</sup> edition.
3. Yuwono Martadinata., (2016). *Arduno itu Pintar* ,PT Elex Media Komputindo, Jakarta
4. Hari Santoso (2015). *Panduan Praktis Arduino untuk Pemula*. Elang Sakti. Malang

## **Bab 5. Sensor Temperatur LM35**

### ***Uraian :***

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai mendeteksi temperature ruangan dengan LM35*

### ***Sasaran :***

*Mahasiswa mampu memahami cara kerja, koding program , skematik serta cara menggunakan LM 35 di Arduino*

### **A. Pengertian LM35**

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. LM35 mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.

Meskipun tegangan sensor ini dapat mencapai 30 volt akan tetapi yang diberikan kesensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60  $\mu\text{A}$  hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (self-heating) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0,5 °C pada suhu 25 °C

Sensor LM35 bekerja dengan mengubah besaran suhu menjadi besaran tegangan. Tegangan ideal yang keluar dari LM35 mempunyai perbandingan 100°C setara dengan 1 volt. Sensor ini mempunyai pemanasan diri (self heating) kurang

dari  $0,1^{\circ}\text{C}$ , dapat dioperasikan dengan menggunakan power supply tunggal dan dapat dihubungkan antar muka (interface) rangkaian control yang sangat mudah.

IC LM 35 sebagai sensor suhu yang teliti dan terkemas dalam bentuk Integrated Circuit (IC), dimana output tegangan keluaran sangat linear terhadap perubahan suhu. Sensor ini berfungsi sebagai penguah dari besaran fisis suhu ke besaran tegangan yang memiliki koefisien sebesar  $10\text{ mV}/^{\circ}\text{C}$  yang berarti bahwa kenaikan suhu  $1^{\circ}\text{C}$  maka akan terjadi kenaikan tegangan sebesar  $10\text{ mV}$ .

Adapun spesifikasi yang dimiliki sensor temperature LM35 ini adalah:

1. Terkalibrasi secara langsung dalam derajat celcius.
2. Keluaran linier  $10.0\text{mV}/\text{derajat celcius}$ .
3. Ketelitian dijamin  $0.5$  derajat celcius pada temperatur  $25$  derajat celcius.
4. Bekerja pada temperatur antara  $-50$  sampai  $150$  derajat celcius.
5. Disiapkan sebagai aplikasi remote.
6. Tegangan operasi antara  $4$  sampai  $20\text{V}$
7. Arus drain lebih kecil dari  $60$  mikro Ampere.
8. Pemanasan sendiri sangat rendah yaitu  $0.08$  derajat celcius dalam udara.
9. Secara tipikal ketidak linieran sebesar  $0.25$  derajat celcius.
10. Impedansi output  $0.1\text{ Ohm}$  untuk beban  $1\text{ mA}$ .

IC LM 35 ini tidak memerlukan pengkalibrasian atau penyetelan dari luar karena ketelitiannya sampai lebih kurang seperempat derajat celcius pada temperature ruang. Jangka sensor mulai dari  $-55^{\circ}\text{C}$  sampai dengan  $150^{\circ}\text{C}$ , IC LM35 penggunaannya sangat mudah, difungsikan sebagai kontrol dari indicator tampilan catu daya terbelah. IC LM 35 dapat dialiri arus  $60\text{ }\mu\text{ A}$  dari supplay sehingga panas yang

ditimbulkan sendiri sangat rendah kurang dari  $0^{\circ}\text{C}$  di dalam suhu ruangan.

Untuk mendeteksi suhu digunakan sebuah sensor suhu LM35 yang dapat dikalibrasikan langsung dalam C (celcius).

### **B. Cara Kerja Sensor LM35**

Sensor LM35 bekerja dengan mengubah besaran suhu menjadi besaran tegangan. Tegangan ideal yang keluar dari LM35 mempunyai perbandingan  $100^{\circ}\text{C}$  setara dengan 1 volt. Sensor ini mempunyai pemanasan diri (self heating) kurang dari  $0,1^{\circ}\text{C}$ , dapat dioperasikan dengan menggunakan power supply tunggal dan dapat dihubungkan antar muka (interface) rangkaian control yang sangat mudah. IC LM 35 sebagai sensor suhu yang teliti dan terkemas dalam bentuk Integrated Circuit (IC), dimana output tegangan keluaran sangat linear terhadap perubahan suhu. Sensor ini berfungsi sebagai penguubah dari besaran fisis suhu ke besaran tegangan yang memiliki koefisien sebesar  $10\text{ mV}/^{\circ}\text{C}$  yang berarti bahwa kenaikan suhu  $1^{\circ}\text{C}$  maka akan terjadi kenaikan tegangan sebesar 10 mV.

IC LM 35 ini tidak memerlukan pengkalibrasian atau penyetelan dari luar karena ketelitiannya sampai lebih kurang seperempat derajat celcius pada temperature ruang. Jangka sensor mulai dari  $-55^{\circ}\text{C}$  sampai dengan  $150^{\circ}\text{C}$ , IC LM35 penggunaannya sangat mudah, difungsikan sebagai kontrol dari indicator tampilan catu daya terbelah. IC LM 35 dapat dialiri arus  $60\ \mu\text{A}$  dari supplay sehingga panas yang ditimbulkan sendiri sangat rendah kurang dari  $0^{\circ}\text{C}$  di dalam suhu ruangan. Untuk mendeteksi suhu digunakan sebuah sensor suhu LM35 yang dapat dikalibrasikan langsung dalam C (celcius), LM35 ini difungsikan sebagai basic temperature sensor.



```

}
void loop()
{
 int value = analogRead(inPin); // read the value from the
 sensor
 lcd.setCursor(0,1);
 float millivolts = (value / 1024.0) * 5000;
 float celsius = millivolts / 10;
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("Temp = ");
 lcd.setCursor(8,0);
 lcd.print(celsius,0);
 lcd.write(0xdf);
 lcd.print("C");
 lcd.setCursor(0,1);
 lcd.print("Temp F = ");
 lcd.setCursor(9,1);
 lcd.print((celsius * 9)/5 + 32); //untuk merubah ke F
 lcd.print("F");
 delay(1000);
}

```

#### **D. Evaluasi**

1. Sensor LM35 dapat kan menggunakan pin digital ?
2. Sebutkan sensor temperature selain LM35 !

#### **E. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktutor Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Ofset, Yogyakarta

## Bab 6 . Sensor Cahaya LDR

### *Uraian :*

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai Sensor LDR*

### *Sasaran :*

*Mahasiswa mampu memahami cara kerja , skematik, dan penggunaan sensor cahaya LDR di Arduino*

### LDR (Light Dependent Resistor)



Gambar 17. LDR

#### **A. Pengertian LDR**

Light Dependent Resistor atau disingkat dengan LDR adalah jenis Resistor yang nilai hambatan atau nilai resistansinya tergantung pada intensitas cahaya yang diterimanya. Nilai Hambatan LDR akan menurun pada saat cahaya terang dan nilai Hambatannya akan menjadi tinggi jika dalam kondisi gelap. Dengan kata lain, fungsi LDR (Light Dependent Resistor) adalah untuk menghantarkan arus listrik jika menerima sejumlah intensitas cahaya (Kondisi Terang) dan menghambat arus listrik dalam kondisi gelap.

Naik turunnya nilai Hambatan akan sebanding dengan jumlah cahaya yang diterimanya. Pada umumnya, Nilai Hambatan LDR akan mencapai 200 Kilo Ohm ( $k\Omega$ ) pada kondisi gelap dan menurun menjadi 500 Ohm ( $\Omega$ ) pada Kondisi Cahaya Terang.

LDR (Light Dependent Resistor) yang merupakan Komponen Elektronika peka cahaya ini sering digunakan atau diaplikasikan dalam Rangkaian Elektronika sebagai sensor pada Lampu Penerang Jalan, Lampu Kamar Tidur, Rangkaian Anti Maling, Shutter Kamera, Alarm dan lain sebagainya. LDR memiliki dua karakteristik yaitu Laju recovery dan respon spektral.

Bila sebuah LDR dibawa dari suatu ruangan dengan level kekuatan tertentu ke dalam suatu ruangan yang gelap. Maka bisa kita amati bahwa nilai resistansi dari LDR tidak akan segera berubah resistansinya pada keadaan ruangan gelap tersebut. Namun LDR tersebut hanya akan bisa mencapai harga di kegelapan setelah mengalami selang waktu tertentu. Laju recovery merupakan suatu ukuran praktis dan suatu kenaikan nilai resistansi dalam waktu tertentu. Harga ini ditulis dalam K/detik, untuk LDR tipe arus harganya lebih besar dari 200K/detik (selama 20 menit pertama mulai dari level cahaya 100 lux), kecepatan tersebut akan lebih tinggi pada arah sebaliknya, yaitu pindah dari tempat gelap ke tempat terang yang memerlukan waktu kurang dari 10 ms untuk mencapai resistansi yang sesuai dengan level cahaya 400 lux.

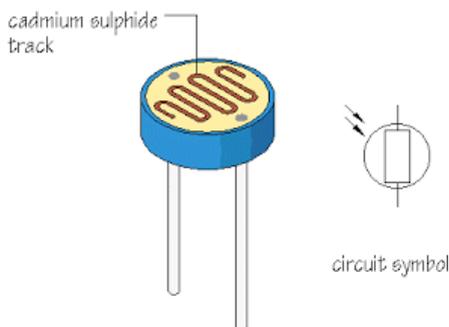
### **Respon Spektral**

LDR tidak mempunyai sensitivitas yang sama untuk setiap panjang gelombang cahaya yang jatuh padanya (yaitu warna). Bahan yang biasa digunakan sebagai penghantar arus listrik yaitu tembaga, aluminium, baja, emas dan perak. Dari kelima bahan tersebut tembaga merupakan penghantar yang paling banyak digunakan karena mempunyai daya hantar yang baik. Resistansi LDR akan berubah seiring dengan perubahan intensitas cahaya yang mengenainya atau yang

ada disekitarnya. Dalam keadaan gelap resistansi LDR sekitar  $10M\Omega$  dan dalam keadaan terang sebesar  $1K\Omega$  atau kurang. LDR terbuat dari bahan semikonduktor seperti kadmium sulfida. Dengan bahan ini energi dari cahaya yang jatuh menyebabkan lebih banyak muatan yang dilepas atau arus listrik meningkat. Artinya resistansi bahan telah mengalami penurunan.

## **B. Prinsip Kerja LDR**

Pada dasarnya LDR terbuat dari sebuah cakram semikonduktor yang mempunyai dua buah elektroda pada permukaannya. Pada saat gelap atau intensitas cahaya rendah, bahan tersebut menghasilkan elektron bebas dengan jumlah yang relatif kecil. Sehingga hanya sedikit elektron yang dihasilkan untuk mengangkut muatan elektrik. Hal ini berarti, pada saat keadaan gelap atau intensitas cahaya rendah, maka LDR akan menjadi konduktor yang buruk, sehingga LDR memiliki resistansi yang besar pada saat gelap atau intensitas cahaya rendah. Pada saat terang atau intensitas cahaya tinggi, bahan tersebut lebih banyak menghasilkan elektron yang lepas dari atom. Sehingga akan lebih banyak elektron yang dihasilkan untuk mengangkut muatan elektrik. Hal ini berarti, pada saat terang atau intensitas cahaya tinggi, maka LDR menjadi konduktor yang baik, sehingga LDR memiliki resistansi yang kecil pada saat terang atau intensitas cahaya tinggi



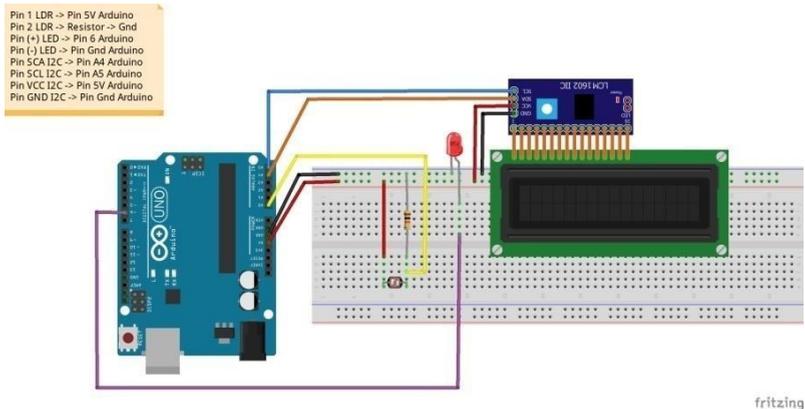
Gambar 18. Bagian dari LDR

### C. Contoh Penggunaan LDR

Pada rangkaian dibawah ini, Sensor LDR berfungsi untuk menangkap cahaya yang ada pada taman. Ketika siang hari LDR menangkap cahaya dari matahari yang menyebabkan output LED tidak menyala. Sebaliknya ketika malam hari LDR tidak menangkap cahaya dan menyebabkan output LED menjadi nyala.

Berikut adalah komponen yang dibutuhkan :

- Arduino Uno R3
- Sensor LDR
- LED
- Resistor 10K
- LCD 16x2
- I2C



Gambar 19. Skematik LDR dan LCD serta LED

### Script

Sketch Sensor LDR pada Arduino Uno

```
/**LAMPU OTOMATIS DENGAN SENSOR CAHAYA
(LDR)***/
```

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);

int hasilSensorLDR; // Variable untuk sensor LDR

void setup() {
 lcd.begin(); //LCD untuk ukuran 16x2
 pinMode(6,OUTPUT); // Set pin 6 sebagai Output
}

void loop() {
 hasilSensorLDR=analogRead(0); // Hasil LDR = Hasil
input pada pin A0
 if (hasilSensorLDR<10) // Jika hasil LDR kurang dari 10
(Kurang Cahaya)
 {
 digitalWrite(6,HIGH); // Aktifkan Relay atau LED
 lcd.setCursor(3, 0);
 lcd.print("SENSOR LDR");
 lcd.setCursor(2, 1); //baris pertama
 lcd.print(" NILAI : ");
 lcd.print(hasilSensorLDR);
 delay(1000);
 lcd.clear();
 }
 else

```

```
digitalWrite(6,LOW); // Jika tidak, Matikan Relay/LED
lcd.setCursor(3, 0);
lcd.print("SENSOR LDR");
lcd.setCursor(2, 1); //baris pertama
lcd.print(" NILAI : ");
lcd.print(hasilSensorLDR);
delay(1000); // Delay setengah detik
lcd.clear();
}
```

#### **D. Evaluasi**

1. Jelaskan Karakteristik LDR terhadap Cahaya?
2. Sensor apa saja yg bisa di gunakan untuk mengukur intensitas cahaya selain LDR ?

#### **E. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktuator Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Offset, Yogyakarta

## **Bab 7. Sensor pendeteksi Manusia PIR**

### ***Uraian :***

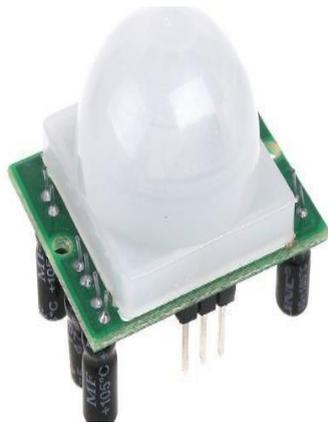
*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai karakteristik dan penggunaan Sensor PIR*

### ***Sasaran :***

*Mahasiswa mampu memahami karakter , skematik , koding dan penggunaan sensor PIR di Arduino .*

### **A. Pengertian PIR**

Sensor PIR (Passive Infrared Receiver) berfungsi untuk menemukan dan menentukan ada tidaknya gerakan manusia. Sebuah sensor yang berbasis inframerah, pada sensor ini tidak terdapat Infrared LED dan fototransistor. PIR bersifat pasif, Sensor ini tidak dapat memancarkan seperti IR LED, PIR hanya bisa memberi tanggapan pancaran dari inframerah yang dimiliki oleh suatu benda.



Gambar 20. PIR

Sensor ini memiliki bagian-bagian yang mempunyai perannya masing-masing, diantaranya Fresnel Lens, IR Filter, Pyroelectric sensor, amplifier dan comparatort. Sensor ini dapat berfungsi dengan cara menangkap/mengumpulkan panas yang mana dapat dihasilkan dari sinar inframerah setiap benda dengan suhu benda diatas nol. Sebagai halnya seorang manusia yang bersuhu tubuh sekitar 32 derajat celcius yang merupakan suhu panas pada lingkungan. Pancaran sinar inframerah kemudian akan diambil oleh Pyroelectric sensor, ini adalah bagian atau komponen utama dari sensor PIR. Pyroelectric sensor terdiri dari galium nitrida, caesium nitrat dan litium tantale yang menghasilkan listrik. Pancaran Inframerah ini mempunyai kekuatan panas oleh karena itu sensor ini dapat menghasilkan arus listrik.

Sensor PIR hanya merespon manusia karena terdapat InfraRed Filter atau penyaring yang menyaring panjang gelombang sinar IR antara 8 - 14 mikrometer. Sehingga panjangnya gelombang yang dapat dihasilkan dari manusia berkisar 9 sampai 10 mikrometer. Dan ini yang dapat dideteksi oleh sensor PIR. Jadi ketika seseorang atau manusia melewati sensor PIR, PIR akan menangkap sinar

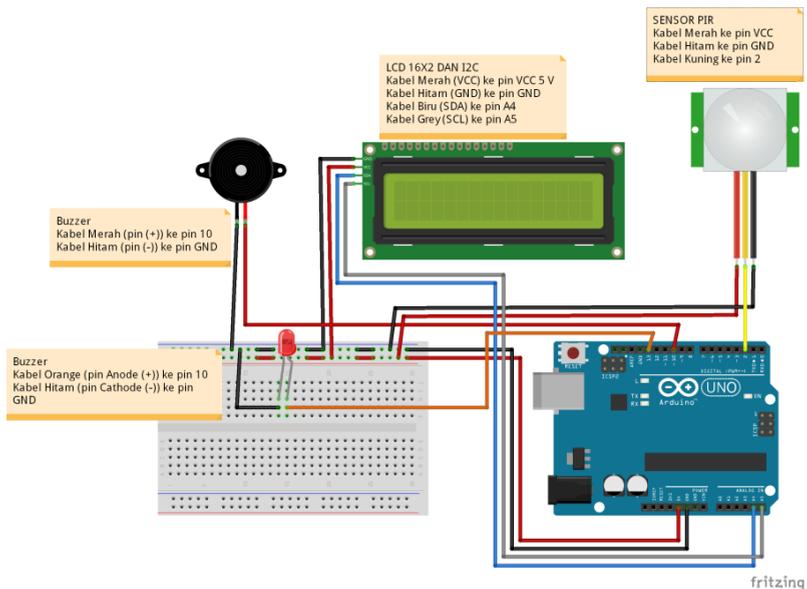
inframerah yang dipancarkan suhu tubuh manusia yang berbeda dari lingkungan sekitar.

Sehingga menyebabkan pyroelectric menghasilkan arus listrik dengan adanya sumber panas oleh sinar inframerah . Kemudian sirkuit amplifier akan menguatkan arus tersebut kemudian akan dibandingkan oleh comparator sehingga menghasilkan output. Sensor PIR memiliki tiga pin, yaitu: VCC dihubungkan ke ground. GND dihubungkan ke ground. OUT dihubungkan ke pemantau keluaran sensor.

## B. Spesifikasi



### C. Contoh Penggunaan PIR



Gambar 22. Skematik PIR, LED, Buzzer dan LCD

#### Script

```
// library lcdi2c
#include <LCD.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

int ledmerah = 13; // Memilih pin indikator LED merah

int inputPir = 2; // Memilih input PIR Sensor

int pirState = LOW; // Pada saat mulai, Matikan sensor
```

```

int val = 0; // Variable pembaca
status pin

int pinbuzzer = 10; // Memilih pin buzzer, dengan pin PWM
LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3,POSITIVE);
void setup()
{
 // ukuran lcd
 lcd.begin(16,2);
 // kondisi awal kosong ketika lcd menyala
 lcd.print("");

 pinMode(ledmerah, OUTPUT); // jadikan LED
 sebagai Output

 pinMode(inputPir, INPUT); // jadikan Sensor
 sebagai Input

 pinMode(pinbuzzer, OUTPUT); // jadikan buzzer
 sebagai output

 Serial.begin(9600);
}

void loop(){
 val = digitalRead(inputPir);
 if (val == HIGH) {

```

```
digitalWrite(ledmerah, HIGH); // led merah menyala
playTone(300, 160); // mengatur bunyi nada buzzer
```

```
delay(150); // jeda waktu bunyi buzzer
```

```
if (pirState == LOW) {
```

```
 lcd.setCursor(0,0);
```

```
 lcd.print("awas ada maling"); // menampilkan pesan pada
 lcd ketika ada pergerakan
```

```
 Serial.println("awas ada maling"); // menampilkan pesan
 pada serial monitor
```

```
 pirState = HIGH;
```

```
 }
```

```
} else {
```

```
 digitalWrite(ledmerah, LOW); // led merah mati
```

```
 playTone(0,0);
```

```
 delay(300);
```

```
 if (pirState == HIGH){
```

```
 lcd.setCursor(0,1);
```

```
 lcd.clear();
```

```
 pirState = LOW; } }
```

```
}
```

```
void playTone(long duration, int freq) {
```

```
 duration *= 1000; // durasi bunyi buzzer
```

```

int period = (1.0 / freq) * 1000000; //frekuensi nada buzzer
long elapsed_time = 0;

while (elapsed_time < duration) {
 digitalWrite(pinbuzzer,HIGH); // buzzer menyala
 delayMicroseconds(period / 2);
 digitalWrite(pinbuzzer, LOW); // buzzer mati
 delayMicroseconds(period / 2);
 elapsed_time += (period);
}
}

```

#### **D. Evaluasi**

1. Jelaskan Cara Kerja Sensor PIR !
2. Dapatkah Sensor PIR mendeteksi Benda Mati ?Kenapa !

#### **F. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktuator Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Offset, Yogyakarta

## **BAB 8. SENSOR ULTRASONIK HCSR04**

### ***Uraian :***

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai karakteristik dan penggunaan Sensor HCSR04*

### ***Sasaran :***

*Mahasiswa mampu memahami karakteristik, skematik , koding dan penggunaan sensor ultrasonic HCSR04 di Arduino*

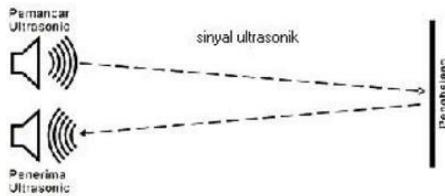
### **A. Pengertian Sensor Ultrasonik**

Sensor ultrasonik adalah suatu sensor yang berfungsi untuk mengukur jarak suatu benda dengan cara kerja yaitu trigger pin memantulkan suatu gelombang dan diterima oleh echo pin sehingga diketahuilah jarak suatu benda . Gelombang sensor ultrasonic menggunakan frekuensi yang sangat tinggi yaitu 20.000 Hz , sehingga bunyi gelombang ultrasonic tidak dapat didengarkan oleh manusia tetapi dapat didengan oleh hewan diantaranya anjing, kucing, kelelawar maupun lumba-lumba

### **B. Cara kerja sensor ultrasonik**

- Sinyal gelombang akan dipantulkan oleh trigger pin dalam rentang waktu tertentu dan frekuensi tertentu, umumnya fkuensi yang digunakan untuk memantulkan suatu gelombang adalah 40KHz
- Gelombang yang dipantulkan akan merambat menjadi gelombang bunyi dengan kecepatan 340 m/s, ketika menyentuh suatu benda maka gelombang akan dipantulkan kembali oleh benda tersebut
- Kemudian gelombang dari hasil pantulan benda tersebut akan diterima oleh echo pin, sehingga akan diketahui waktu tempuh yang dibutuhkan untuk memantulkan suatu gelombang
- Dari waktu pantul gelombang tersebut akan

dikonversikan oleh program di arduino menjadi satuan cm dengan rumus jarak= (durasi/2) \*0.03483



Gambar 22. Arah pancar ultrasonik

Sensor ultrasonic yang dapat mengukur jarak benda mulai dari 2cm sampai 4m, sensor ini banyak digunakan para pecinta mikrokontroller dikarenakan harganya yang relative murah. Sensor HC04 mempunyai 4 buah pin, yaitu vcc pin, ground, echo pin, dan trigger pin.

Untuk mengukur jarak suatu benda dengan satuan cm pada sensor HC04 menggunakan rumus “ JARAK = (WAKTU/2)\*0.03483 atau “JARAK=(WAKTU/2)/29.1” , dimana waktu adalah lama pantulan gelombang ultrasonic yang dipantulkan ke sebuah benda dengan proses : TRIGER > BENDA > ECHO, sehingga perlu dibagi 2 untuk mengetahui jarak dari ultrasonic ke benda tanpa kembali ke echo

$$c_{\text{air}} = (331.3 + 0.606 \cdot \vartheta) \text{ m/s,}$$

Dimana *teta* adalah suhu udara, maka

$$c = 331.5 + 0.606 * [\text{suhu udara}]$$

Jika Suhu 28 per celcius.

$$c = 331.5 + 0.606 * 28$$

$$c = 348.3 \text{ meter/second}$$

Dikarenakan distance mempunyai satuan mili/s. maka convert 348.3 m/s ke micr/s. Hasilnya  $348.3 * / 1000 = 0.03483$  micro/s, sehingga didapatlah suatu rumus :

$$\text{JARAK} = (\text{WAKTU}/2)*0.03483$$



- Vcc pin berfungsi sebagai tegangan positif 5v
- Ground sebagai tegangan negative untuk catu daya listrik
- Trigger pin berfungsi untuk memantulkan suatu gelombang dengan rentang waktu dan frekuensi tertentu
- Echo pin berfungsi untuk menerima pancaran gelombang yang dipantulkan oleh trigger pin, sehingga akan diketahui jarak dari suatu benda.

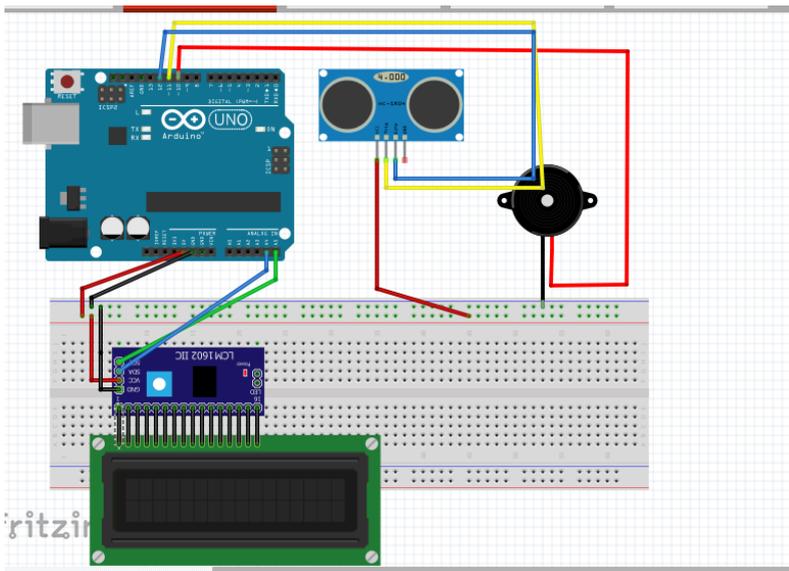
### **C. Contoh Penggunaan HC-SR04**

Alur skematik :

- Trigger pin sensor ultrasonic pada pin 11 memantulkan suatu gelombang pantul untuk kemudian gelombang pantul tersebut diterima oleh echo pin pada pin 12 arduino sehingga menghasilkan suatu nilai pantul, kemudian pada program arduino nilai pantul tersebut dikonversikan menjadi nilai jarak

dengan satuan cm.

- Nilai jarak tersebut akan ditampilkan kedalam lcd 16x2 dengan menggunakan module i2c melalui pin SDA ke pin A4 arduino dan pin SCL ke pin A5 arduino
- Berdasarkan sketch program pada arduino, apabila nilai jarak lebih dari 20 cm maka speaker akan berbunyi melalui pin 10, dan jika nilai jarak  $< 20$  cm maka speaker akan off



Gambar 24.Skematik HC-SR04,LCDI2C dan Buzzer

### Script

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16,
2);

int trig= 11;
int echo= 12;
```

```

int speaker = 10;
long durasi,
jarak;

void setup() {

 pinMode(trig, OUTPUT);
 pinMode(echo, INPUT);
 pinMode(speaker,
 OUTPUT); lcd.begin ();

 lcd.setCursor(1, 0);
 lcd.print("SINAR
 COMPUTER");

 lcd.setCursor(3, 1);
 lcd.print("Ultrasonic
 "); delay(3000);
 Serial.begin(9600);

}

void loop() {

 lcd.clear();
 lcd.setCursor(1,0)
; lcd.print("Jarak :
 ");
 lcd.print(jarak);
 lcd.print(" cm");
 lcd.setCursor(2,1)
;

 digitalWrite(trig,
 LOW);
 delayMicroseconds(8);

```

```
digitalWrite(trig,
HIGH);
delayMicroseconds(8);
digitalWrite(trig,
LOW);
delayMicroseconds(8);
```

```
durasi= pulseIn(echo,
HIGH); jarak= (durasi/2) /
29.1; Serial.println(jarak);
delay(5);
```

```
if (jarak < 20) {
 lcd.print(" Awas nabrak ");
 digitalWrite(speaker,
 HIGH); delay(500);
} else if (jarak > 20){
 lcd.print(" jarak aman ");
 digitalWrite(speaker,
 LOW);
```

```

 delay(500);
}
delay(1000);
// lcd.clear();

}

```

Keterangan untuk script diatas :

- **#include <Wire.h>**  
**#include <LiquidCrystal\_I2C.h>**  
**LiquidCrystal\_I2C lcd(0x3F, 16,**  
**2);**

Perintah untuk menambahkan library “**Wire.h**” dan “**LiquidCrystal\_I2C.h**”, kemudian pada baris ketiga adalah untuk menentukan alamat dan jenis lcd yang digunakan

- **int trig=**  
**11; int**  
**echo= 12;**  
**int speaker = 10;**  
**long durasi,**  
**jarak;**

Untuk membuat suatu variable trig, echo , dan speaker dengan masing-masing pin sesuai diatas, pada baris keempat digunakan untuk membuat variable jarak dan durasi yang akan digunakan untuk rumus ultrasonic

- **void setup() {**  
**pinMode(trig,**

```
OUTPUT);
pinMode(echo,
INPUT);
pinMode(speaker,
OUTPUT); lcd.begin ();
```

Ini adalah proses konfigurasi awal dimana untuk inialisasi pin yang digunakan

```
 lcd.setCursor(1, 0);
lcd.print("SINAR
COMPUTER");
lcd.setCursor(3, 1);
lcd.print("Ultrasonic ");
```

```
delay(3000);
Serial.begin(9600
);
}
```

Perintah untuk menampilkan karakter di lcd, pada baris pertama akan menampilkan karakter “SINAR COMPUTER” dan pada baris kedua akan menampilkan karakter “ULTRASONIC” dengan delay 3000 milisecond

```
□ void loop() {
 lcd.clear();
 lcd.setCursor(1,0);
 lcd.print("Jarak :
"); lcd.print(jarak);
 lcd.print(" cm");
 lcd.setCursor(2,1);
}
```

Ini adalah tahap konfigurasi atau program utama, pada script diatas lcd akan menampilkan karakter Jarak (nilai) cm pada baris pertama

```
□ digitalWrite(trig,
 LOW);
 delayMicroseconds(8);
 digitalWrite(trig,
 HIGH);
 delayMicroseconds(8);
 digitalWrite(trig,
 LOW);
 delayMicroseconds(8);
```

ini adalah konfigurasi pin Trigger pada ultrasonic, dimana pada kondisi awal trigger pin harus dalam kondisi low ,

kemudian Trigger pin diaktifkan dengan delay 8 Microsecond untuk memantulkan sebuah gelombang bunyi

```
□ durasi= pulseIn(echo,
HIGH); jarak= (durasi/2)
/ 29.1;
Serial.println(jarak);
delay(5);
```

scrip diatas digunakan untuk menghitung jarak ultrasonic dengan suatu benda, dimana “pulseIn” adalah rentang waktu yang dibutuhkan echo pin untuk berpindah dari kondisi LOW ke HIGH,

kemudian dari durasi(waktu pantul ) tadi diubah oleh variable jarak menjadi satuan centimeter

```
□ if (jarak < 20) {
lcd.print(" Awas nabrak
"); digitalWrite(speaker,
HIGH); delay(500);

} else if (jarak > 20){
lcd.print(" jarak aman ");
digitalWrite(speaker,
LOW); delay(500);

}
```

Ini adalah suatu perintah utama dimana ketika jarak ultrasonic kurang dari 20 cm maka speaker melalui pin 10 akan menyala, begitu pula sebaliknya jika jarak lebih dari 20 cm maka speaker akan mati.

#### **D. Evaluasi**

1. Dapatkan Sensor HCSR04 menembus dinding ?  
Kenapa?

2. Selain sensor HCSR04 sensor apa saja yang dapat di gunakan untuk mengukur jarak ?

#### **E. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktuator Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Ofset, Yogyakarta

## **Bab 9 . Sensor Pengukur Kelembapan Tanah ( Soil Moisture)**

### ***Uraian :***

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai Sensor Moisture untuk mengukur kelembapan tanah*

### ***Sasaran :***

*Mahasiswa mampu memahami karakteristik , skematik , koding dan penggunaan sensor soil moisture di arduino*

### **A. Pengertian Soil Moisture**

Sensor kelembapan tanah mengukur volumetrik kadar air di tanah. Sejak langsung pengukuran gravimetri kelembapan tanah bebas membutuhkan menghapus, pengeringan, dan bobot sampel, sensor kelembapan tanah mengukur kadar air volumetrik secara tidak langsung dengan menggunakan beberapa properti lain dari tanah, seperti hambatan listrik, konstanta dielektrik, atau interaksi dengan neutron , sebagai proxy untuk kadar air. Hubungan antara diukur properti dan tanah air harus dikalibrasi dan dapat bervariasi tergantung pada factor lingkungan seperti jenis tanah, suhu , atau konduktivitas listrik . Tercermin microwave radiasi dipengaruhi oleh kelembapan tanah dan digunakan untuk penginderaan jauh di hidrologi dan pertanian. Instrumen probe portabel bisa digunakan oleh petani atau tukang kebun.

Sensor kelembapan tanah biasanya mengacu pada sensor yang memperkirakan kadar air volumetrik. Kelas lain dari sensor mengukur properti lain dari kelembapan di tanah yang disebut potensial air ; sensor ini biasanya disebut sebagai

potensi sensor air tanah dan termasuk tensiometer dan blok gipsum.

Sensor kelembaban ini bisa membaca jumlah kelembaban yang ada di tanah yang mengelilinginya. Ini adalah sensor berteknologi rendah, tapi ideal untuk memantau taman kota, atau tingkat air tanaman hewan peliharaan Anda. Ini adalah harus memiliki alat untuk kebun terhubung.

Sensor ini menggunakan dua probe untuk melewati arus melalui tanah, dan kemudian dibaca bahwa ketahanan untuk mendapatkan tingkat kelembaban. Lebih banyak air membuat tanah melakukan listrik lebih mudah (kurang tahan), sedangkan tanah kering melakukan listrik dengan buruk (lebih tahan).

Akan sangat membantu jika Anda mengingatkan Anda untuk menyirami tanaman indoor Anda atau untuk memantau kelembaban tanah di kebun Anda.

## **B. Spesifikasi**

- Catu daya: 3.3v atau 5v
- Sinyal voltase keluaran: 0 ~ 4.2v
- Saat ini: 35mA
- Definisi Pin:
  1. Output analog (kawat biru)
  2. GND (kawat hitam)
  3. Daya (kawat merah)
- Ukuran: 60x20x5mm
- Rentang nilai:
  - a) 0 ~ 300: tanah kering
  - b) 300 ~ 700: tanah lembab
  - c) 700 ~ 950: dalam air

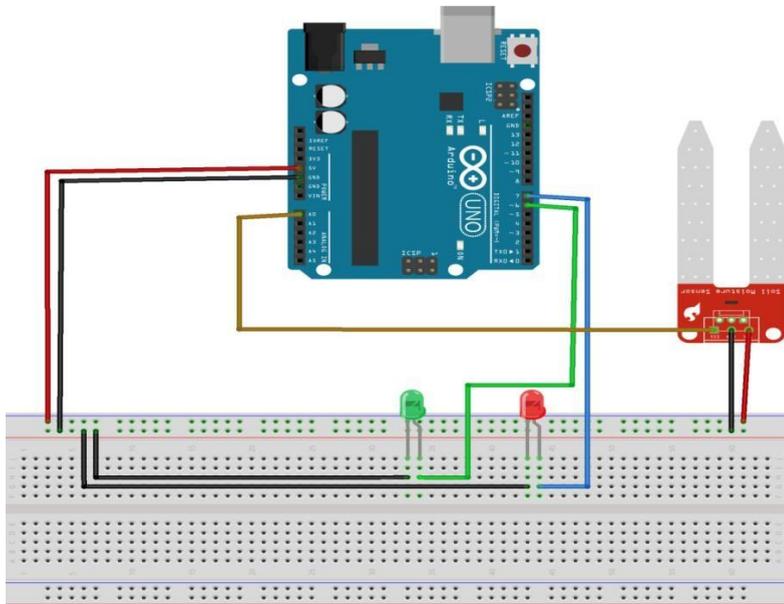
## **Teknologi**

Teknologi yang biasa digunakan untuk secara tidak langsung mengukur kadar air volumetrik (kelembaban tanah) termasuk)

- Frekuensi Domain Reflectometry (FDR) : The konstanta dielektrik dari elemen volume tertentu di sekitar sensor diperoleh dengan mengukur frekuensi operasi dari sebuah rangkaian berosilasi.
- Waktu Domain Transmisi (TDT) dan Time Domain Reflectometry (TDR) : The konstanta dielektrik dari elemen volume tertentu di sekitar sensor diperoleh dengan mengukur kecepatan propagasi di sepanjang saluran transmisi dimakamkan.
- Pengukur Neutron kelembaban : The moderator sifat air untuk neutron yang digunakan untuk memperkirakan kadar air tanah antara sumber dan detektor penyelidikan.
- Tanah resistivitas : Mengukur seberapa kuat tanah menolak aliran listrik antara dua elektroda dapat digunakan untuk menentukan kadar air tanah.

- Galvanic cell : Jumlah air yang dapat ditentukan berdasarkan tegangan tanah menghasilkan karena air bertindak sebagai elektrolit dan menghasilkan listrik. Teknologi di balik konsep ini adalah sel galvanik.

### C. Cara Penggunaan Soil Moisture



Gambar 24. Skematik Kelembapan tanah dan LED

#### Script

```
int kl = A0;//membuat variabel Soil Moisture pada pin A0
Arduino int ijo = 6;//membuat variabel Led hijau pada pin 6
Arduino
```

```
int abang = 7;//membuat variabel Led merah pada pin 7
Arduino int batas = 800;//membuat variabel untuk batas
sensor Soil Moisture
```

```
void setup(){
```

```
Serial. begin(9600); //komunikasi serial dari
Arduino ke Komputer
```

```
pinMode(kl, INPUT); //mengatur Sensor Soil Moisture
menjadi Input
```

```
pinMode(ijo, OUTPUT); //mengatur LED Hijau menjadi
Output pinMode(abang, OUTPUT); //mengatur LED Merah
menjadi Output
```

```
digitalWrite(ijo, LOW); //menyatakan bahwa LED Hijau
dalam kondisi mati
```

```
digitalWrite(abang, LOW); //menyatakan bahwa LED merah
dalam kondisi mati
```

```
}
```

```
void loop() {
```

```
 // membaca masukkan pada pin analog 0
```

```
 int sensorValue = analogRead(kl); //membuat variabel
sensor Soil Moisture
```

```
 Serial. print(sensorValue); //untuk memunculkan nilai
dari Soil Moisture
```

```
 if(sensorValue < batas){ //jika nilai dari Soil Moisture
lebih kecil dari bata
```

```
 Serial. println(" - Tanah Lembab, banyune kakehan"
```

```
); //menampilkan pesan pada serial monitor
 digitalWrite(abang, LOW); //kondisi led merah
 mati digitalWrite(ijo, HIGH); //kondisi led hijau
 menyala
```

```
}
```

```
else {
```

```
Serial.println(" - Tanah Garing, banyu mu endi");//menampilkan pesan pada serial monitor

digitalWrite(abang, HIGH);//kondisi led merah menyala
digitalWrite(ijo, LOW);//kondisi led hijau mati
}

delay(500);// memberi waktu jeda 500 milidetik
}
```

#### **D. Evaluasi**

1. Bagaimana cara kerja sensor Soil Moisture?
2. Dapat kan sensor Moisture mendeteksi Minyak ?

#### **E. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktutor Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Ofset, Yogyakarta

## **Bab 10. Motor Servo**

### ***Uraian :***

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai Motor Servo*

### ***Sasaran :***

*Mahasiswa mampu memahami karakteristik , skematik , koding dan penggunaan motor servo di arduino*

## **Motor Servo**



Gambar 25.Motor Servo SG90

### **A. Pengertian Motor Servo**

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor. motor servo merupakan perangkat yang terdiri

dari motor DC, serangkaian gear, rangkaian kontrol dan potensiometer.

Serangkaian gear yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo.

Servo biasanya digunakan untuk aplikasi robotic, radio control, bahkan untuk pengaturan mekanikal pada pesawat terbang. Hal ini dikarenakan Sistem kontrol servo yang paling cocok untuk kecepatan tinggi, aplikasi torsi tinggi yang melibatkan perubahan beban dinamis. Untuk alternatifnya, Sistem kontrol motor Stepper yang lebih murah dan optimal untuk aplikasi yang membutuhkan akselerasi rendah-menengah, torsi memegang tinggi, dan fleksibilitas operasi loop terbuka atau tertutup.

## **B. Macam-macam motor servo**

1. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri.
2. servo DC biasanya lebih cocok untuk digunakan pada aplikasi-aplikasi yang lebih kecil.

Dan bila dibedakan menurut rotasinya, umumnya terdapat dua jenis motor servo yang terdapat di pasaran, yaitu motor servo rotation  $180^\circ$  dan servo rotation continuous.

## **C. Spesifikasi Motor Servo SG90**

- Dimensi (p x l x t) = 22 x 11,5 x 27 mm
- Berat bersih = 9 gram (10,6 gram bila kabel dan konektor ikut ditimbang)
- Torsi maksimum / stall torque = 1,2 kg.cm
- Rentang sudut putaran  $180^\circ$
- Catu daya operasional = 4 ~ 7,2 Volt DC
- Operating Voltage = 4.0 to 7.2 volts

- Kecepatan pada 4,8 VDC tanpa beban = 0,12 detik per 60° (57,6 rpm)
- *Dead band* = 10 $\mu$ s
- Kecepatan operasi pada 4.8V (kondisi tanpa beban) = 0.12 sec/ 60 degrees
- Panjang kabel = 248 mm
- Connector Wire Length = 9.75 inches (248 mm)
- Tipe konektor: universal "S" (Futaba / JR / Berg / dll)
- Bahan gir plastic

#### D. Komponen Motor Servo



Gambar 26. Komponen Motor Servo

Motor servo pada dasarnya dibuat menggunakan motor DC yang dilengkapi dengan controler dan sensor posisi sehingga dapat memiliki gerakan 0°, 90°, 180° atau 360°. Berikut adalah komponen internal sebuah motor servo 180°.

Tiap komponen pada motor servo diatas masing-masing memiliki fungsi sebagai controler, driver, sensor, girbox dan aktuator. Pada gambar diatas terlihat beberapa bagian komponen motor servo. Motor pada sebuah motor

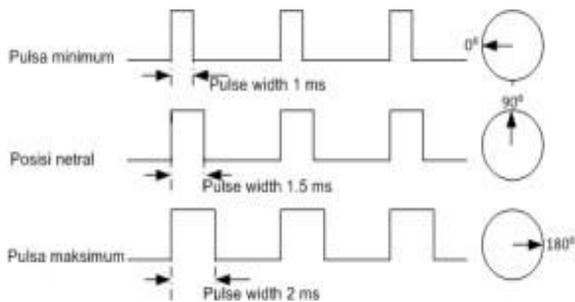
servo adalah motor DC yang dikendalikan oleh bagian controller, kemudian komponen yang berfungsi sebagai sensor adalah potensiometer yang terhubung pada sistem gearbox pada motor servo.

### **Pin Pada Motor Servo**

- Kabel warna merah merupakan kabel power yang berfungsi untuk mengkoneksikan dengan tegangan 5V pada board arduino.
- Kabel yang berwarna coklat/hitam merupakan kabel ground yang nantinya akan dihubungkan dengan ground yang ada pada board arduino.
- Kabel kuning yang merupakan kabel pin signal servo yang akan dihubungkan dengan pin input pada board arduino.

### **E. Prinsip Kerja Motor Servo**

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (Pulse Wide Modulation / PWM) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Sebagai contoh, lebar pulsa dengan waktu 1,5 ms (mili detik) akan memutar poros motor servo ke posisi sudut 90°. Bila pulsa lebih pendek dari 1,5 ms maka akan berputar ke arah posisi 0° atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1,5 ms maka poros motor servo akan berputar ke arah posisi 180° atau ke kanan (searah jarum jam). Lebih jelasnya perhatikan gambar dibawah ini.



Gambar 26. Posisi sudut motor servo

Ketika lebar pulsa kendali telah diberikan, maka poros motor servo akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (rating torsi servo). Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms (mili detik) untuk menginstruksikan agar posisi poros motor servo tetap bertahan pada posisinya.

## **F. Cara Mengendalikan Motor Servo**

Untuk menjalankan atau mengendalikan motor servo berbeda dengan motor DC. Karena untuk mengendalikan motor servo perlu diberikan sumber tegangan dan sinyal kontrol. Besarnya sumber tegangan tergantung dari spesifikasi motor servo yang digunakan. Sedangkan untuk mengendalikan putaran motor servo dilakukan dengan mengirimkan pulsa kontrol dengan frekuensi 50 Hz dengan periode 20ms dan duty cycle yang berbeda. Dimana untuk menggerakkan motor servo sebesar 90° diperlukan pulsa

dengan ton duty cycle pulsa positif 1,5ms dan untuk bergerak sebesar  $180^\circ$  diperlukan lebar pulsa 2ms.

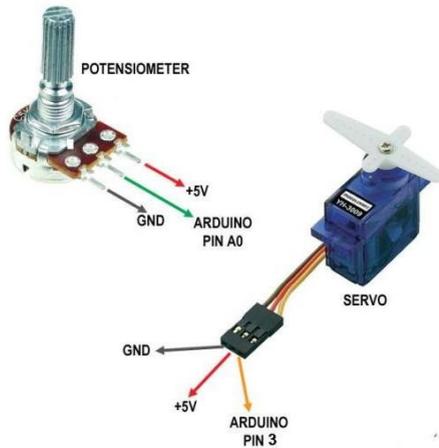
### **Kelebihan dan Kekurangan Motor Servo**

- a. Kelebihan
  - Tidak bergetar dan tidak ber-resonansi saat beroperasi.
  - Daya yang dihasilkan sebanding dengan ukuran dan berat motor.
  - Penggunaan arus listrik sebanding dengan beban yang diberikan.
  - Resolusi dan akurasi dapat diubah dengan hanya mengganti encoder yang dipakai.
  - Tidak berisik saat beroperasi dengan kecepatan tinggi.
- b. Kekurangan
  - Memerlukan pengaturan yang tepat untuk menstabilkan umpan balik.
  - Motor menjadi tidak terkendali jika encoder tidak memberikan umpan balik.

Beban berlebih dalam waktu yang lama dapat merusak motor

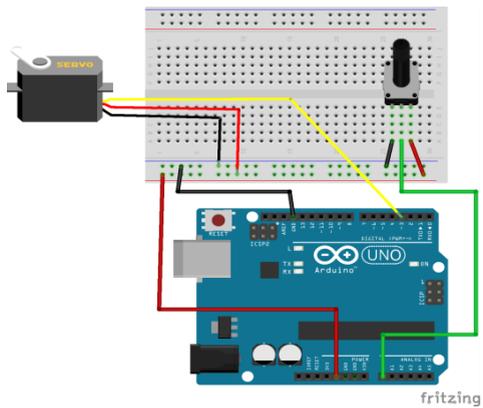
Pada kasus ini menggunakan potensio untuk mengatur perputaran dari servo.

## Komponen Potensio dan Servo



**Gambar 27. Komponen Servo dan Potensio**

### G. Contoh Penggunaan Motor Servo



**Gambar 28. Skematik Rangkaian**

## Script

```
#include <Servo.h> //menyertakan library
servo pada program

Servo servoku; //membuat variabel
servoku untuk servo

int pot= A0; //membuat variabel pot
untuk sensor pin A0

int nilaipot= 0; //membuat variabel
nilaipot untuk menyimpan nilai potensio

void setup(){
 servoku.attach(3); //memilih pin 3 digital
 untuk servoku
}

void loop(){
 nilaipot= analogRead(pot); //menyimpan nilai
 pot pada variabel nilaipot

 nilaipot= map(nilaipot, 0, 1023, 10, 180); //konversi nilai 0-
 1023 menjadi 10-180

 servoku.write(nilaipot); //menggerakan
 servo berdasarkan nilaipot
}
```

Keterangan :

1. **nilaipot= analogRead(pot);**  
menyimpan nilai dari potensio pada variabel nilai pot
2. **nilaipot= map(nilaipot, 0, 1023, 10, 180);** mengkonversi nilai potensio dari 0-1023 menjadi 10-180, ini menyesuaikan dengan jenis servo yang saya gunakan, yaitu hanya bisa sampai maksimal 180 derajat. Dan saya set dari 10 bukan dari 0, karena untuk servo ini jika diprogram kurang 10 derajat akan bergetar dan menyebabkan servo cepat panas.

## H. Evaluasi

1. Dapatkah Servo berputar  $360^0$  ? jelaskan !
2. Perintah apa yang di gunakan untuk mengkontrol servo pada pin digital ?

## I. Referensi

1. Dian Artanto, (2017). *Interface Sensor dan Aktutor Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Offset, Yogyakarta

## Bab 11. Sensor Alkohol MQ-3

### *Uraian :*

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai sensor alcohol MQ-3 untuk mendeteksi Alkohol*

### *Sasaran :*

*Mahasiswa mampu memahami karakteristik, sekmatik, koding dan penggunaan MQ-3 dalam Arduino*

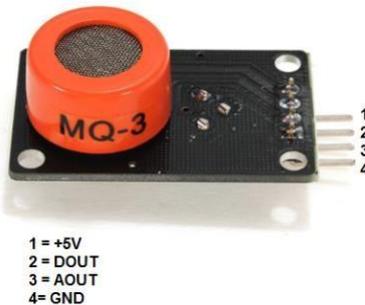


**Gambar 29. Sensor MQ 3**

### **A. Pengertian MQ-3**

**MQ-3** adalah sensor alkohol berbasis pemanas yang mengeluarkan sinyal analog (biasanya diinterpretasikan di suatu tempat antara 150 dan 1023 tergantung pada berapa lama Anda membiarkan sensornya memanaskan), yang melalui penggunaan kode dan kalibrasi Arduino Anda, dapat ditafsirkan. Untuk apa pun yang Anda butuhkan. (<http://bildr.org/2013/10/mq3-arduino>)

Sensor gas MQ-3 ini digunakan untuk mengukur konsentrasi gas atau uap Alkohol di udara menggunakan rangkaian mikrokontroler seperti Arduino. Sensor MQ-3 ini sangat sensitif terhadap gas Alkohol. Material gas yang dideteksi oleh sensor gas MQ3 adalah gas atau uap Alkohol.



**Gambar 30. Pin dalam MQ 3**

## **B. Spesifikasi**

- Sensitivitas tinggi dengan area deteksi luas
- Long life
- Detection gas : Alkohol
- Concentration : 0.05- 10 mg/L
- Circuit Voltage( $V_c$ ) : 5V
- Heating Voltage ( $V_h$ ) : 1.4V-5V
- Heating Time  $T_h$  (High) : 60s
- Heating Time  $T_h$  (Low) : 90s
- Load Resistance ( $R_L$ ) : 200K
- Heater resistance ( $R_h$ ) : 33 ohm
- Heater Consumption : <750 mW
- Sensing resistance : 1M ohm - 8M ohm (pada 0.4mG/L Alkohol)
- Preheat time : >24 jam

## **C. Cara kerja**

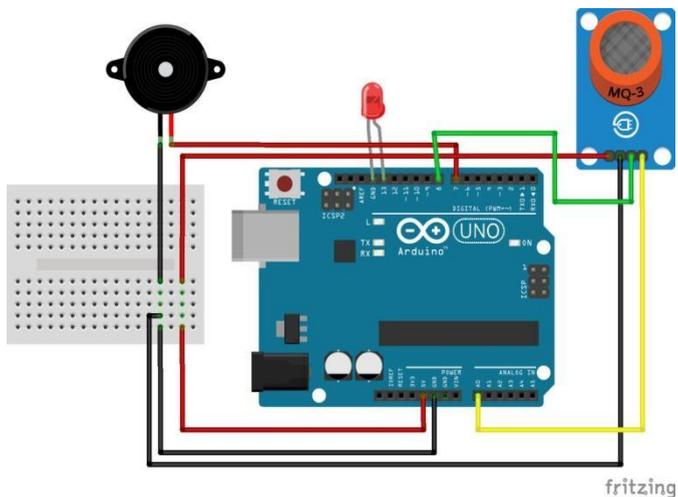
Sensor bekerja melalui terminal AOOUT memberikan output tegangan analog sebanding dengan jumlah alkohol yang dideteksi sensor. Semakin banyak alkohol yang dideteksi, semakin besar tegangan analog yang akan dihasilkannya. Sebaliknya, semakin sedikit alkohol yang

dideteksi, semakin sedikit tegangan analog yang dihasilkannya. Jika tegangan analog mencapai ambang tertentu, maka akan mengirimkan pin digital DOUT yang tinggi. Setelah pin DOUT ini naik tinggi, arduino akan mendeteksi hal ini dan akan memicu LED menyala, menandakan bahwa ambang alkohol telah tercapai dan sekarang melampaui batas.

#### A. KELEBIHAN DAN KEKURANGAN

- Kelebihan dan
  - Kepekaan terhadap alkohol yang tinggi dan rendah terhadap bensin
  - Waktu respon cepat
  - Stabil dan tahan lama
  - Sumber tegangan AC / Dc 5 Volt
  - Suhu Operasional – 10 s/d 70 derajat celcius
- Kekurangan
  - Mengonsumsi daya yang cukup besar

#### D. Penggunaan MQ-3



**Gambar 31.** Skematik Pendeteksi Alkohol

### Script

```
// deklarasi Sensor MQ 3
#define ledPin 13 // memanggil Led pin
int gasPin = 8; // deklarasi Sensor MQ-3
int value = 0;
#define buzzer 7 // memanggil buzzer
const int threshold = 400; // Kapasitas kadar Alkohol yang
terdeteksi
void setup() {
Serial.begin(9600);
pinMode(ledPin,OUTPUT);
pinMode(buzzer,OUTPUT);
}

void loop() {
 value = analogRead(0);
 // jika terdeteksi oleh sensor MQ-3 maka :
 if (value > threshold) {
 Serial.print ("Kadar Alkohol : "); // menampilkan kadar
 alkohol yang terdeteksi
 digitalWrite(buzzer, HIGH); // posisi buzzer menyala
 Serial.println(value,DEC);
 digitalWrite(ledPin, LOW); // posisi led mati
 }
 else { // posisi Sensor MQ-3 stanbay
 Serial.print("Kadar Alkohol normal : "); // menampilkan
 posisi normal
 Serial.println(value,DEC);
 digitalWrite(buzzer, LOW); // posisi buzzer mati
 digitalWrite(ledPin, HIGH); // posisi led menyala
 }

 delay(1000);
```

### **E. Evaluasi**

1. Sebutkan Cara kerja Dari MQ 3
2. Dapat kah MQ 3 mendeteksi Debu ?Jelaskan !

### **F. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktutor Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Offset, Yogyakarta

## **Bab 12. Sensor MQ-7 pendeteksi Karbon**

### **Uraian :**

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai MQ-7 untuk mendeteksi MQ-7*

### **Sasaran :**

*Mahasiswa mampu memahami dan mengerti karakterestik, koding, skematik dan penggunaan MQ-7 untuk mendeteksi karbon.*

### **A. Pengertian MQ-7**

MQ 7 merupakan sensor gas yang digunakan dalam peralatan untuk mendeteksi gas karbon monoksida (CO) dalam kehidupan sehari-hari, industri, atau mobil. Fitur dari sensor gas MQ7 ini adalah mempunyai sensitivitas yang tinggi terhadap karbon monoksida (CO), stabil, dan berumur panjang. Sensor ini menggunakan catu daya heater : 5V AC/DC dan menggunakan catu daya rangkaian : 5VDC, jarak pengukuran : 20 - 2000ppm untuk ampu mengukur gas karbon monoksida.

#### **Kondisi Standar Sensor Bekerja**

- VC/(Tegangan Rangkaian) =  $5V \pm 0.1$
- VH (H)/ Tegangan Pemanas (Tinggi) =  $5V \pm 0.1$
- VH (L)/ Tegangan Pemanas (Rendah) =  $1.4V \pm 0.1$
- RL/Resistansi Beban Dapat disesuaikan
- RH Resistansi Pemanas =  $33\Omega \pm 5\%$
- TH (H) Waktu Pemanasan (Tinggi) =  $60 \pm 1$  seconds
- TH (L) Waktu Pemanasan (Rendah) =  $90 \pm 1$  seconds
- PH Konsumsi Pemanasan = Sekitar 350mW

#### **Kondisi Lingkungan**

- Tao/Suhu Penggunaan =  $-20^{\circ}\text{C} - 50^{\circ}\text{C}$

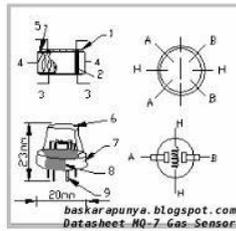
- Tas/Suhu Penyimpanan =  $-20^{\circ}\text{C}$ - $50^{\circ}\text{C}$
- RH/Kelembapan Relatif = kurang dari 95%RH
- O<sub>2</sub> Konsentrasi Oksigen = 21%(stand condition)  
(Konsentrasi Oksigen dapat mempengaruhi sensitivitas)

#### Karakteristik Sensitivitas

- Rs/ Tahanan Permukaan Terhadap Tubuh = 2-20k pada 100ppm Carbon Monoxide(CO)
- $a(300/100\text{ppm})$ / Tingkat Konsentrasi Kemiringan = Kurang dari 0.5 Rs (300ppm)/Rs(100ppm)
- Standar Kondisi Bekerja = Temperature  $-20^{\circ}\text{C}\pm 2^{\circ}\text{C}$   
Kelembapan  $65\%\pm 5\%$  , RL: $10\text{K}\Omega\pm 5\%$ , Vc: $5\text{V}\pm 0.1\text{V}$   
VH: $5\text{V}\pm 0.1\text{V}$ , VH: $1.4\text{V}\pm 0.1\text{V}$
- Waktu Panaskan Tidak kurang dari 48 jam
- Jarak Deteksi: 20ppm-2000ppm carbon monoxide

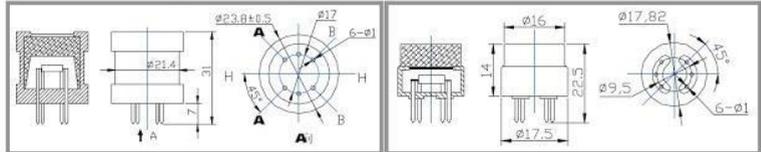
### **B. Struktur dan Konfigurasi MQ-7**

Struktur dan konfigurasi MQ-7 sensor gas ditunjukkan pada gambar. 30 (Konfigurasi A atau B), sensor disusun oleh mikro AL<sub>2</sub>O<sub>3</sub> tabung keramik, Tin Dioksida (SnO<sub>2</sub>) lapisan sensitif, elektroda pengukuran dan pemanas adalah tetap menjadi kerak yang dibuat oleh plastik dan stainless steel bersih. Pemanas menyediakan kondisi kerja yang diperlukan untuk pekerjaan komponen sensitif. MQ-7 dibuat dengan 6 pin, 4 dari mereka yang digunakan untuk mengambil sinyal, dan 2 lainnya digunakan untuk menyediakan arus pemanasan.



| Parts                    | Materials                                       |
|--------------------------|-------------------------------------------------|
| 1 Gas sensing layer      | SnO <sub>2</sub>                                |
| 2 Electrode              | Au                                              |
| 3 Electrode line         | Pt                                              |
| 4 Heater coil            | Ni-Cr alloy                                     |
| 5 Tubular ceramic        | Al <sub>2</sub> O <sub>3</sub>                  |
| 6 Anti-explosion network | Stainless steel gauze network (SUS316 100-mesh) |
| 7 Clamp ring             | Copper plating Ni                               |
| 8 Resin base             | Bakelite                                        |
| 9 Tube Pin               | Copper plating Ni                               |

Fig.1



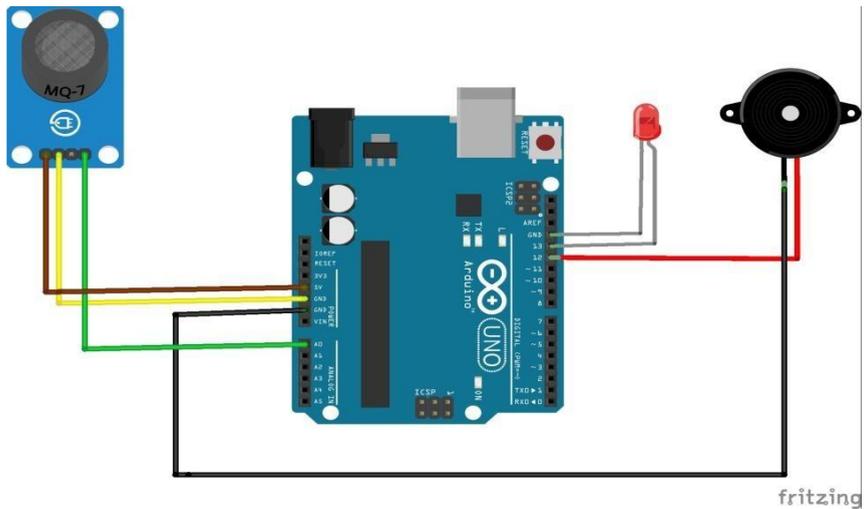
Gambar 30.Komponen MQ7

### C. Contoh Penggunaan MQ 7

#### Skematik yang dibutuhkan :

1. 1 buah Arduino UNO
2. 1 buah Sensor MQ7
3. 1 buah BUZZER
4. 1 buah LED
5. Beberapa kabel Jumper

## SKEMATIK



**Gambar 32.Skematik MQ7 ,Buzzer dan Led**

### Script

```
// ----- Project Alat Pendeteksi Gas CO ----- //
// ----- Berbasis Arduino dan Sensor Gas MQ-7 -----
//
/*
 Untuk menentukan kapan asap terdeteksi, maka lihat data
 pada serial monitor
*/
//inialisasi pin 90ariab
const int pinAnalog = A0;
//inialisasi pin alarm
const int pinAlarm = 12;
//inialisasi pin led
int led = 13;
//inialisasi 90ariable data
int data;
```

```

// ----- program stting/default awal----- //
void setup()
{
 //inisialisasi baud rate serial monitor
 Serial.begin(9600);
 //inisialisasi status I/O pin
 pinMode(pinAnalog, INPUT);
 pinMode(pinAlarm, OUTPUT);
 pinMode(led, OUTPUT);
}

// ----- Program Looping/Berulang Terus-Menerus ----- //
void loop()
{
 //data adalah hasil dari pembacaan sensor
 data = analogRead(pinAnalog);

 //menulis pada serial monitor nilai dari data
 Serial.println(data);
 //delay pembacaan dalam satuan milidetik
 delay(300);
 //ubah nilai data ini sesuai dengan nilai yang keluar
 //pada serial monitor saat sensor berada dekat asap
 //.....//
 //jika nilai data lebih besar atau sama dengan 300
 if (data >= 300)
 {
 //alarm dibunyikan dan led hidup
 digitalWrite(pinAlarm, HIGH);
 digitalWrite(led, HIGH);
 delay(500);
 digitalWrite(pinAlarm, LOW);
 digitalWrite(led, LOW);
 delay(500);
 }

 //ubah nilai data ini sesuai dengan nilai yang keluar
 //pada serial monitor saat sensor tidak mendeteksi asap

```

```
//.....//
//jika tidak sesuai dengan ketentuan di atas
else
{
//alarm dimatikan
digitalWrite(pinAlarm, LOW);
}
}
.
```

#### **D. Evaluasi**

1. Dapat kah Sensor MQ-7 mendeteksi Gas LPG ?
2. Apa perbedaan MQ 3 dan MQ 7 ?

#### **E. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktuator Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Offset, Yogyakarta

## Bab 13. Sensor Kelembaban Udara dan Temperatur DHT11

### *Uraian :*

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai DHT 11*

### *Sasaran :*

*Mahasiswa mampu memahami pengertian, skematik, koding dan penerapan pengukuran kelembaban dengan DHT 11 di Arduino*



Gambar 33. Sensor Kelembaban Udara dan Temperatur Dht11

### A. Pengertian sensor kelembaban

Suatu alat ukur yang digunakan untuk membantu dalam proses pengukuran atau pendefinisian yang suatu kelembaban uap air yang terkandung dalam udara.

Jenis kelembaban

1. Kelembaban absolut adalah bilangan yang menunjukkan berapa gram uap air yang tertampung dalam satu meter kubik udara
2. Kelembaban relative adalah bilangan yang

menunjukkan berapa persen perbandingan antara uap air yang ada dalam udara saat pengukuran dan jumlah uap air maksimum yang dapat ditampung oleh udara tersebut.

## **SENSOR KELEMBABAN DHT 11**

DHT11 adalah sensor digital yang dapat mengukur suhu dan kelembaban udara di sekitarnya. Sensor ini sangat mudah digunakan bersama dengan Arduino. Memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang akurat. Koefisien kalibrasi disimpan dalam OTP program memory, sehingga ketika internal sensor mendeteksi sesuatu, maka module ini menyertakan koefisien tersebut dalam kalkulasinya.

DHT11 termasuk sensor yang memiliki kualitas baik, dinilai dari respon, pembacaan data yang cepat, dan kemampuan anti-interference. Ukurannya yang kecil, dan dengan transmisi sinyal hingga 20 meter, membuat produk ini cocok digunakan untuk banyak aplikasi-aplikasi pengukuran suhu dan kelembaban.

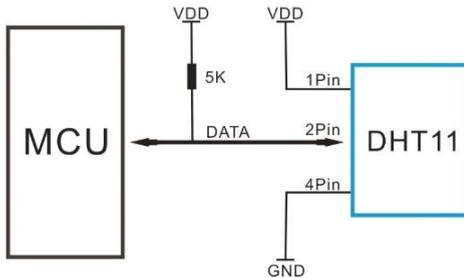
### **B. SPESIFIKASI**

- Supply Voltage: +5 V
- Temperature range : 0-50 °C error of  $\pm 2$  °C
- Humidity : 20-90% RH  $\pm 5$  % RH error
- Interface : Digital

| Parameters                     | Conditions               | Minimum | Typical     | Maximum |
|--------------------------------|--------------------------|---------|-------------|---------|
| <b>Humidity</b>                |                          |         |             |         |
| <b>Resolution</b>              |                          | 1%RH    | 1%RH        | 1%RH    |
|                                |                          |         | 8 Bit       |         |
| <b>Repeatability</b>           |                          |         | ± 1%RH      |         |
| <b>Accuracy</b>                | 25 °C                    |         | ± 4%RH      |         |
|                                | 0-50 °C                  |         |             | ± 5%RH  |
| <b>Interchangeability</b>      | Fully interchangeable    |         |             |         |
| <b>Measurement Range</b>       | 0 °C                     | 30%RH   |             | 90%RH   |
|                                | 25 °C                    | 20%RH   |             | 90%RH   |
|                                | 50 °C                    | 20%RH   |             | 80%RH   |
| <b>Response Time (Seconds)</b> | 1/e(63%)25 °C , 1m/s Air | 6 S     | 10 S        | 15 S    |
| <b>Hysteresis</b>              |                          |         | ± 1%RH      |         |
| <b>Long-Term Stability</b>     | Typical                  |         | ± 1%RH/year |         |
| <b>Temperature</b>             |                          |         |             |         |
| <b>Resolution</b>              |                          | 1 °C    | 1 °C        | 1 °C    |
|                                |                          | 8 Bit   | 8 Bit       | 8 Bit   |
| <b>Repeatability</b>           |                          |         | ± 1 °C      |         |
| <b>Accuracy</b>                |                          | ± 1 °C  |             | ± 2 °C  |
| <b>Measurement Range</b>       |                          | 0 °C    |             | 50 °C   |
| <b>Response Time (Seconds)</b> | 1/e(63%)                 | 6 S     |             | 30 S    |

Gambar 34. Spesifikasi DHT 11

### SKEMA INTERNAL SENSOR DHT11

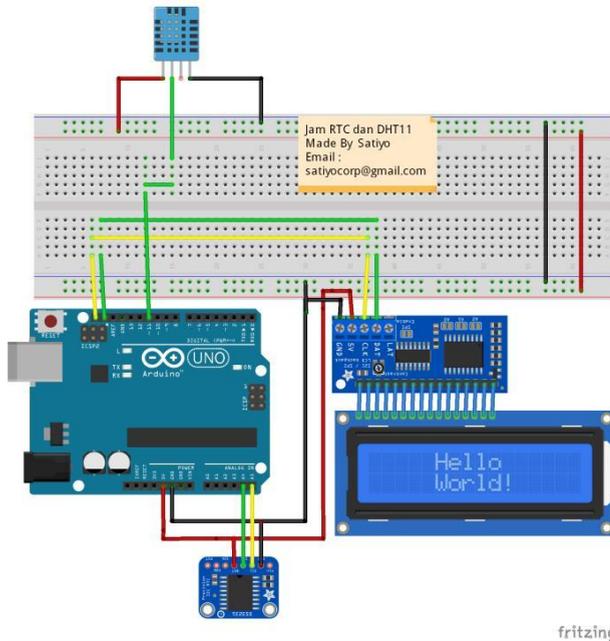


Sensor ini menggunakan komunikasi serial format data single-bus di gunakan untuk berkomunikasi dan sinkronisasi antara sensor DHT11 Dan MCU, proses komunikasi membutuhkan waktu 40ms.

Untuk menggunakan sensor DHT 11 ini ada berbagai cara. Tapi pada kasus ini untuk penggunaan sensor DHT 11 akan menggunakan bantuan library tertentu. Sehingga pada program yang digunakan sudah lebih disederhanakan.

Berikut Contoh skema dan program membuat jam digital sekaligus menampilkan suhu dan kelembaban pada lcd

### C. Contoh pemakaian Dht 11 dan Rtc



### SKETCH

```
//Pemanggilan library Wire, RTC, dan LCD
#include <SimpleTimer.h> // library pengganti delay (
multitask)
#include <Wire.h>
#include <DS3231.h>
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x3F ,16,2);
```

```
////////// variable DHT ////////////
```

```
#include <DHT.h>
```

```
#define DHTPIN 11
```

```
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
float humidity, temp_f; // Values read from sensor
```

```
//int ledPins[]={4,5,6,7,8,9};
```

```
SimpleTimer timer;
```

```
// Inisialisasi RTC DS3231
```

```
DS3231 rtc(SDA, SCL);
```

```
Time t; //pencacah string time()
```

```
uint8_t hh = 0, mm = 0, ss = 0, dd = 0, bb = 0;
```

```
int yy = 0;
```

```
String Day = " ";
```

```
String Month = " ";
```

```
//mengganti angka hari ke kar akrer
```

```
char* chday[] =
```

```
{"Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jum'at",
"Sabtu" };
```

```
//mengganti angka bulan ke karakrer
```

```
const char* chbul[] =
```

```
{"Dec", "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
"Aug", "Sep", "Oct", "Nov" };
```

```
byte termometer[8] = //membuat icon / simbol termometer
```

```
{
```

```
 B00100,
```

```
 B01010,
```

```
 B01010,
```

```
 B01110,
```

```
 B01110,
```

```
B11111,
B11111,
B01110 };
```

```
byte humidit[8] = //membuat icon / simbol kelembaban
{
 B00100,
 B00100,
 B01010,
 B01010,
 B10001,
 B10001,
 B10001,
 B01110,
};
```

```
byte customC[8] = //membuat huruf c kecil si atas
{
 0b01110,
 0b10000,
 0b10000,
 0b10000,
 0b01110,
 0b00000,
 0b00000,
 0b00000
};
```

```
int i;
```

```
void setup ()
```

```
{
 //pinMode (0, OUTPUT); //Untuk Fungsi Alarm
 Wire.begin(); // memulai koneksi i2c
 lcd.init(); // inialisasi lcd 12c
 lcd.backlight(); // menhidupkan backlight lcd i2c
 rtc.begin(); //Memulai RTC
```

```

Serial.begin(9600); // memulai koneksi serial dengan baud
rate 9600
delay(10); //

if (rtc.begin(),0)
{
Serial.println("RTC Tidak Tersedia");
lcd.print("RTC Tidak Tersedia");
while (1);
}

//Gunakan untuk setting RTC pertama kali (setelah RTC di
setup upload ulang tanpa kode set RTC)
//rtc.setTime (18, 35, 00); //set jam (hh,mm,ss)
//rtc.setDate(14, 5, 2017); //set tanggal (dd,bb,yyyy)

// Fungsi simple timer, mengatur interval program di
jalankan
timer.setInterval(10000, tampilJam);
timer.setInterval(10000, tampilDHT);
//timer.setInterval(60000, ledJalan);

//Menampilkan Runningtext Pada StartUp
lcd.clear();
lcd.begin(16,2);
lcd.setCursor(16,0);
lcd.print(" SATIYO ");
lcd.setCursor(16,1);
lcd.print(" ARIHYOSHI ");
for (i = 0 ; i < 16; i ++) {
lcd.scrollDisplayLeft();
delay(200);}
delay(2000);
lcd.clear();

```

```

// memanggil dan menampilkan kode byte simbol yang
sudah di buat pada lcd
lcd.createChar(1,termometer);
lcd.createChar(2,humidit);
lcd.createChar(0,customC);
}

////script led berJalan ////

/*for (int i =0; i <6; i++)
{
 pinMode(ledPins[i],OUTPUT);
}*/

////////// Program Jam Digital //////////
void tampilJam()
{
 //Menuliskan jam & tanggal pada lcd
 t = rtc.getTime(); //Mengambil Data Jam
 Day = rtc.getDOWStr(FORMAT_SHORT); //mengambil
data String nama Hari
 Month = rtc.getMonthStr(1); //mengambil data String
nama Bulan
 hh = t.hour,DEC;
 mm = t.min,DEC;
 ss = t.sec,DEC;
 dd = t.date,DEC;
 bb = t.mon,DEC;
 yy = t.year,DEC;

//Menampilkan Jam pada lcd
lcd.setCursor(0, 0);
if (hh<10){lcd.print(F("0"));}
lcd.print(hh);
lcd.print(":");
if(mm<10){lcd.print(F("0"));}

```

```
lcd.print(mm);
```

```
//menampilkan tanggal pada lcd
lcd.setCursor(7, 0);
if (dd<10){lcd.print(F(" "));}
lcd.print(dd);
lcd.setCursor(9, 0);
lcd.print(" ");
lcd.print(bb[chbul]);
lcd.print(" ");
lcd.print((yy)-2000);
}
```

```
////////// Program sensor dht11 ////////////
```

```
void tampilDHT()
```

```
{
float hum = dht.readHumidity(); // membuat variable
mengambil data kelembaban dht11
float temp = dht.readTemperature(); //membuat variable
mengambil data suhu
```

```
// menampilkan suhu pada lcd
```

```
lcd.setCursor(1, 1);
lcd.write(1);
lcd.setCursor(3, 1);
lcd.print((float)temp,0);
lcd.setCursor(5, 1);
lcd.write(0xdf);
lcd.write((uint8_t)0);
```

```
//menampilkan kelembaban pada lcd
```

```
lcd.setCursor(9, 1);
lcd.write(2);
lcd.setCursor(11, 1);
lcd.print((float)hum,0);
lcd.print("%");
```

```

 delay(1000);
}
////////// Program Led Berjalan //////////
/*void ledJalan()
{
 for(int i =0; i<6; i++)
 {
 digitalWrite(ledPins[i],HIGH);
 delay(100);
 digitalWrite(ledPins[i],LOW);
 }

 for(int i =5; i<1; i--)
 {
 digitalWrite(ledPins[i],HIGH);
 delay(100);
 digitalWrite(ledPins[i],LOW);
 }
}*/

void loop ()
{
 timer.run(); //menjalankan simple timer
 tampilJam(); // menjalankan program penampil jam
 tampilDHT(); // menjalankan program penampil suhu dan
 kelembaban
 //ledJalan();
}

```

Cara kerja program di atas ,data dari RTC DS3221 dan sensor DHT11 di konversi ke string melalui library yang sudah tersedia, dan di panggil dengan program arduino, yang nantinya data tersebut di tampilkan melalui layar LCD 16x2.

#### **D. Evaluasi**

1. Apa perbedaan DHT 11 dan DHT 22 ?
2. Dapat kah DHT 11 menggunakan Pin digital ?

#### **E. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktutor Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Ofset, Yogyakarta

## **Bab 14. Sensor Suara**

### ***Uraian :***

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai sensor suara*

### ***Sasaran :***

*Mahasiswa mampu memahami karakteristik , skematik, koding dan penggunaan sensor suara di Arduino*

### **A. Pengertian Sensor Suara**

Sensor suara adalah alat sensor yang mampu mengubah gelombang Sinusioda suara menjadi gelombang sinus energi listrik (Alternating Sinusioda Electric Curret). Sensor suara bekerja berdasarkan besar kecilnya kekuatan gelombang suara yang diterima membran sensor dan menyebabkan Bergeraknya membran sensor dimana juga terdapat sebuah kumparan dibalik membran tadi yang naik turun mengikuti besar kecilnya penerimaan gelombang suara.

### **Prinsip Kerja Sensor Suara**

Cara kerja sensor suara adalah dengan merubah besaran suara menjadi besaran listrik. Sinyal suara yang masuk akan diolah sehingga akan menghasilkan suatu kondisi boolean yaitu kondisi 1 (aktif) atau kondisi 0 (mati), yang mana sinyal kondisi tersebut dibaca oleh mikrokontroler.

Sensor suara adalah sensor yang cara kerjanya merubah besaran gelombang suara menjadi besaran listrik. Komponen yang termasuk dalam sensor suara yaitu :

## ECM

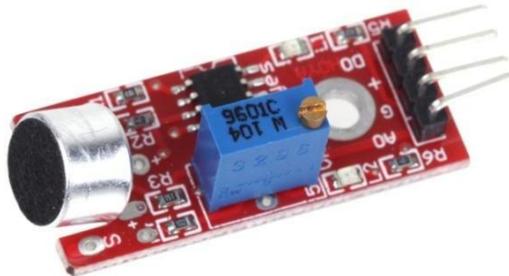
ECM atau Electric Condenser Microphone biasa juga disebut mic kondenser adalah microphone yang terbuat dari lempeng konduktor tipis berbentuk sebuah kapasitor yang dapat berubah-ubah nilai kapasitasnya sesuai dengan getaran suara yang diterima

## Sensor Suara V2

Modul sensor suara memiliki output pin analog dan digital. Modul sensor suara menggunakan input mic condensor. Sensor ini dapat mendeteksi suara dan sebagai deteksi sensor saklar ke arduino atau mikrokontroler.

Spesifikasi Modul :

1. Voltage : 5V
2. Indikator LED
3. Tingkat output TTL
4. Output Analog dan Digital
5. Dilengkapi Dioda Pelindung



Gambar 34 .Sensor Suara

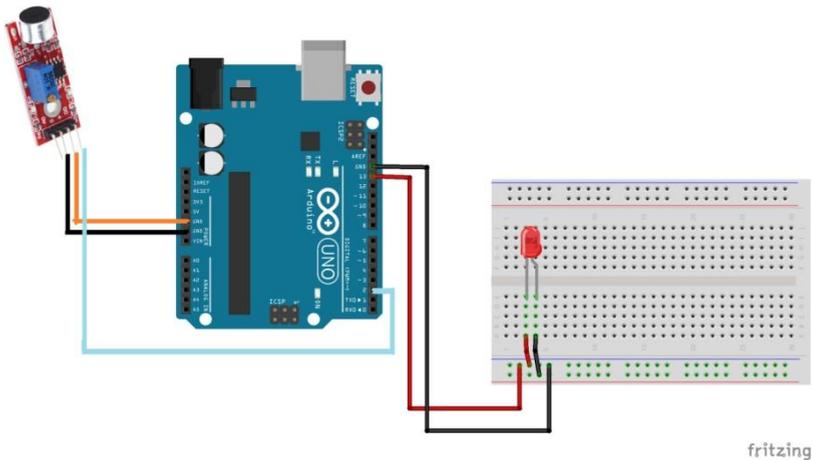
## **B. Contoh Penggunaan Sensor Suara KY-038**

Pada era serba teknologi dan canggih saat ini untuk melakukan banyak hal tidaklah sulit malah mudah dan praktis. Namun masih sedikit aplikasi yang canggih digunakan untuk tempat tinggal atau rumah, jadi untuk mempermudah kegiatan dalam ruangan saya ingin melakukan simulasi menyalakan dan mematikan lampu

menggunakan sensor suara yang dimana tidak perlu repot berjalan ke ujung ruangan untuk mematikan atau menyalakan lampu melalui saklar, tinggal tepuk tangan saja di area yang dijangkau sensor maka lampu akan menyala secara otomatis, dan untuk mematikan lampu tinggal tepuk tangan 1 kali lagi. Yah ini sangat mudah dan praktis sekali !

### **Alat & Bahan**

- Arduino Uno
- Led
- Sensor Suara (KY-038)
- Breadboard
- JumperWiring Arduino



Gambar 35. Skema Pengkabelan arduino led dan sensor suara.

### Script

```
int Led=13; // deklarasi Led pada pin 13
int OutputDO=2; // deklarasi pin 2 sebagai output
sensor
//int val; //val sebagai buffer data
boolean val=0; //kondisi sensor untuk output digital
int lampu; // deklarasi Led pin 13 dinamakan lampu

void setup(){
//Inialisasi I/O
pinMode(Led,OUTPUT);
pinMode(OutputDO,INPUT);
}

void loop(){
lampu=digitalRead(Led);
//Membaca sinyal keluaran dari sensor berupa logika 1
atau 0
val=digitalRead(OutputDO);
```

```

if(val==LOW && lampu==LOW) { //kondisi sensor
suara low dan led/lampu low
//Jika berlogika 1 maka LED akan menyala
digitalWrite(Led,HIGH); //ketika sensor menerima suara
led/lampu akan menyala
//delay(5000);
}

else if (val==LOW && lampu==HIGH)
{digitalWrite(Led,LOW);}

```

### **C. Evaluasi**

1. Dapatkah sensor suara di atur tingkat ke sensitifitas mendeteksi suara nya ?bagaimana ?
2. Adakah jenis sensor suara yang lain nya ?

### **D. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktutor Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Offset, Yogyakarta

## **Bab 15 . Project Helm Pintar Pendeteksi Gas**

### **Uraian :**

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai pembuatan project dan penggabungan berbagai sensor*

### **Sasaran :**

*Mahasiswa mampu memahami pembuatan project arduino*

### **A. Latar Belakang**

Perancangan helm cerdas proyek merupakan suatu sistem keamanan yang dapat memberikan sinyal bahaya ketika kondisi lapangan dalam kondisi tidak baik. Penelitian ini dibuat agar pekerja lebih aman ketika melakukan pekerjaannya. Sistem keamanan ini menggunakan sensor LDR untuk mendeteksi cahaya, sensor MQ-2 yang berfungsi mendeteksi gas, dan sensor DHT11 untuk suhu. Penelitian ini diawali dengan membuat konsep project alat yang dibuat. Dari hasil penelitian, dalam kondisi kurang cahaya senter pada helm akan otomatis hidup, pada MQ-2 ketika gas pada di lapangan terlalu tinggi dan juga pada DHT11 ketika suhu tidak memadai maka otomatis akan memberikan sinyal lampu dan suara.

**A.**

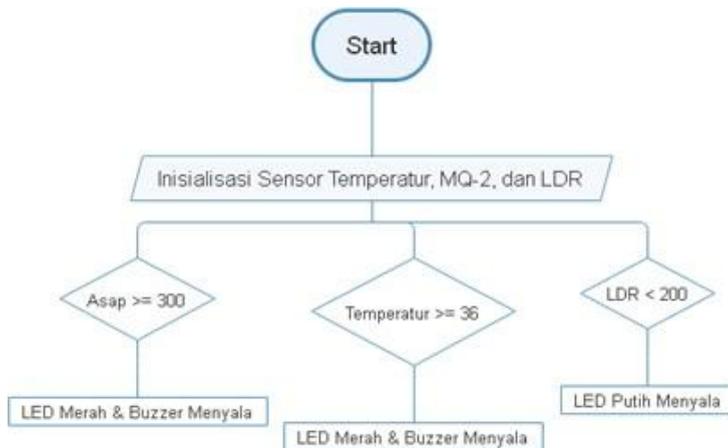
**B.**

**C.**

**D.**

**E.**

## B. Alur Kerja Sistem

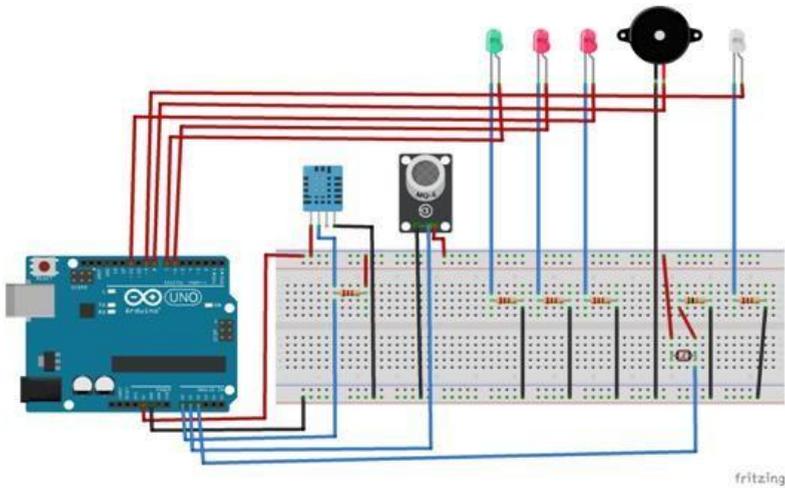


**Gambar 36.alur kerja Helm**

Cara kerja helm ini adalah ketika sensor DHT11 mendeteksi suhu diatas suhu 50 °C maka buzzer dan led merah akan menyala. Ketika MQ-2 mendeteksi asap nilai asap lebih dari 300 ppm maka buzzer dan led merah. Jika kondisi aman, maka led warna hijau akan menyala. Sedangkan sensor LDR akan menyalakan lampu senter jika cahaya kurang.

## C. Project Arduino DIY helm Pintar

### Skematik



Gambar 37. Skematik Helm Cerdas Proyek



Gambar 3.8 Hasil Project

## Script

```
#include <Adafruit_Sensor.h>
#include <DHT.h>

int red = 6;
int red2 = 11;
int green = 7;
int buzzer = 8;
int lamp = 9;
int mq2 = A1;
int Vmq2 = 250;
int ldr = A2;
int Vldr = 200;
#define DHTPIN A0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
 pinMode (red , OUTPUT);
```

```

pinMode (red2 , OUTPUT);
pinMode (green , OUTPUT);
pinMode (lamp , OUTPUT);
pinMode (buzzer , OUTPUT);
pinMode (mq2, INPUT);
pinMode (ldr, INPUT);
Serial.begin(9600);
dht.begin();
}

void loop() {
float t = dht.readTemperature();
int Amq2 = analogRead(mq2);
int Aldr = analogRead(ldr);

if (isnan(t)) {
Serial.println("Failed to read from DHT sensor!");
return;
}

Serial.print("Temperature : ");
Serial.print(t);
Serial.println(" *C ");
Serial.print("Gas Value : ");
Serial.println(Amq2);
Serial.print("Light Value : ");
Serial.println(Aldr);
Serial.println(" ");
delay(2000);

if (t >= 36 && Amq2 > Vmq2 && Aldr < Vldr) {
digitalWrite(red, HIGH);
delay(100);
digitalWrite(red, LOW);
delay(100);
digitalWrite(red2, HIGH);
}
}

```

```
 delay(100);
 digitalWrite(red2, LOW);
 delay(100);
 digitalWrite(green, LOW);
 digitalWrite(lamp, HIGH);
 tone(buzzer, 1000, 200);
}
else if (t < 36 && Amq2 > Vmq2 && Aldr < Vldr)
{
 digitalWrite(red, HIGH);
 delay(100);
 digitalWrite(red, LOW);
 delay(100);
 digitalWrite(red2, HIGH);
 delay(100);
 digitalWrite(red2, LOW);
 delay(100);
 digitalWrite(green, LOW);
 digitalWrite(lamp, HIGH);
 tone(buzzer, 1000, 200);
}
else if (t < 36 && Amq2 > Vmq2 && Aldr > Vldr)
{
 digitalWrite(red, HIGH);
 delay(100);
 digitalWrite(red, LOW);
 delay(100);
 digitalWrite(red2, HIGH);
 delay(100);
 digitalWrite(red2, LOW);
 delay(100);
 digitalWrite(green, LOW);
 digitalWrite(lamp, LOW);
 tone(buzzer, 1000, 200);
}
```

```

else if (t >= 36 && Amq2 < Vmq2 && Aldr >
Vldr) {
 digitalWrite(red, HIGH);
 delay(100);
 digitalWrite(red, LOW);
 delay(100);
 digitalWrite(red2, HIGH);
 delay(100);
 digitalWrite(red2, LOW);
 delay(100);
 digitalWrite(green, LOW);
 digitalWrite(lamp, LOW);
 tone(buzzer, 1000, 200);
}
else if (t > 36 && Amq2 < Vmq2 && Aldr < Vldr)
{
 digitalWrite(red, HIGH);
 delay(100);
 digitalWrite(red, LOW);
 delay(100);
 digitalWrite(red2, HIGH);
 delay(100);
 digitalWrite(red2, LOW);
 delay(100);
 digitalWrite(green, LOW);
 digitalWrite(lamp, HIGH);
 tone(buzzer, 1000, 200);
}
else if (t < 36 && Amq2 < Vmq2 && Aldr < Vldr)
{
 digitalWrite(red, LOW);
 digitalWrite(red2, LOW);
 digitalWrite(green, HIGH);
 digitalWrite(lamp, HIGH);
 noTone(buzzer);
}
}

```

```

else if (t < 36 && Amq2 > Vmq2 && Aldr > Vldr)
{
 digitalWrite(red, HIGH);
 delay(100);
 digitalWrite(red, LOW);
 delay(100);
 digitalWrite(red2, HIGH);
 delay(100);
 digitalWrite(red2, LOW);
 delay(100);
 digitalWrite(green, LOW);
 digitalWrite(lamp, LOW);
 tone(buzzer, 1000, 200);
}
else if (t < 36 && Amq2 < Vmq2 && Aldr > Vldr)
{
 digitalWrite(red, LOW);
 digitalWrite(red2, LOW);
 digitalWrite(green, HIGH);
 digitalWrite(lamp, LOW);
 noTone(buzzer);
}
delay(100);
}

```

#### D. Evaluasi

1. Buatlah project Sederhana dengan DHT 11 dan MQ 2

## **E. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktuator Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Ofset, Yogyakarta

## **Bab 16 . Project Penyiraman Tanaman Otomatis**

### ***Uraian :***

*Mata kuliah ini memberikan pengetahuan kepada mahasiswa mengenai pembuatan project dan penggabungan berbagai sensor*

### ***Sasaran :***

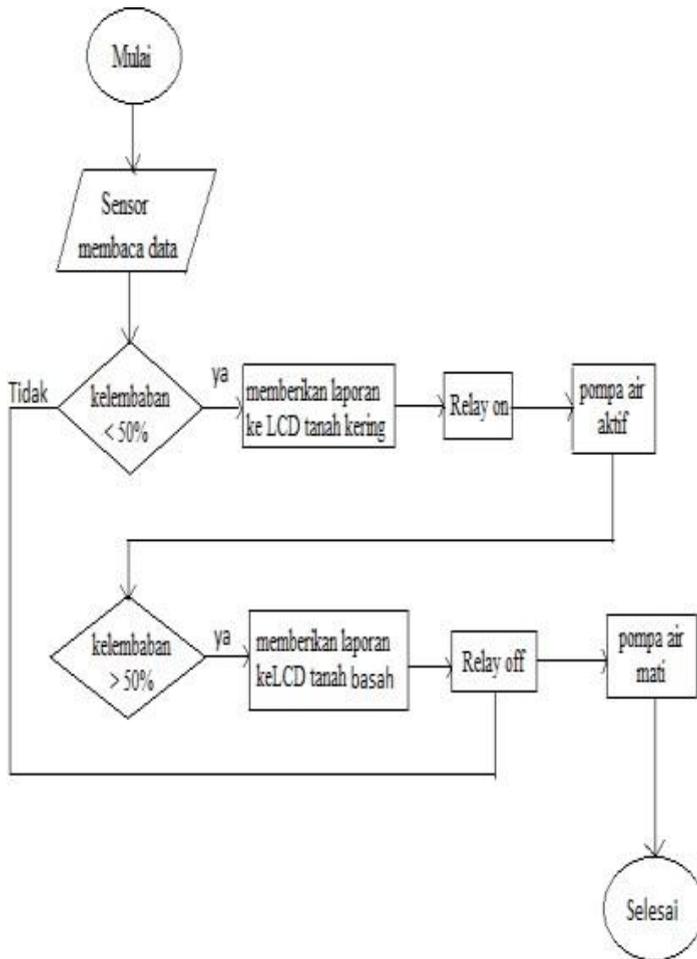
*Mahasiswa mampu memahami pembuatan project arduino*

## **PENYIRAMAN TANAMAN OTOMATIS MENGGUNAKAN SENSOR KELEMBABAN TANAH**

### **A. Latar Belakang**

Pada project kali ini berupa sistem yang mendeteksi kadar air dalam tanah di mana air suatu yang sangat penting bagi tanaman. Project ini menggunakan sensor Soil Moisture. Sistem ini bekerja berdasarkan pendeteksian dari soil moisture sensor, yang kemudian akan ditampilkan pada LCD berupa status kelembabannya menghasilkan output berupa pompa air yang akan menyala jika kondisi tanah dibawah 50 % dan pompa air tersebut akan mati jika kondisi tanah diatas 50%.

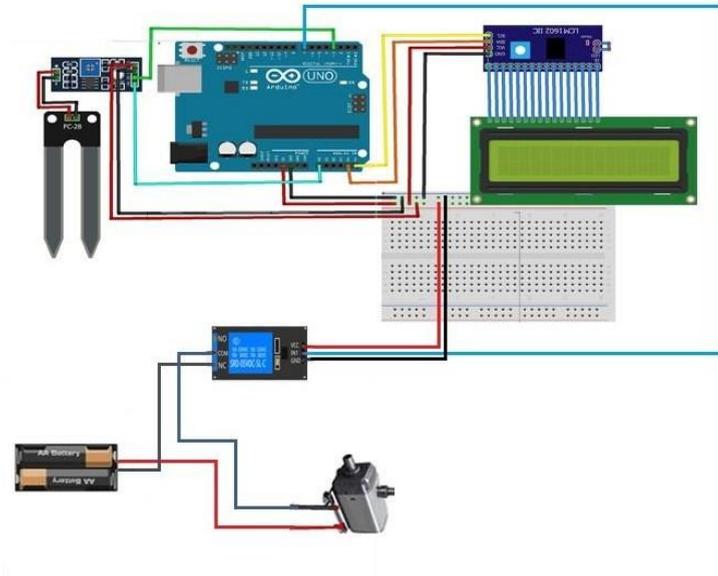
## B. Alur Kerja Sistem



Gambar 3.9 Alur Sistem

## C. Project Penyiraman Otomatis Arduino DIY

### Skematik



Gambar 40 skematik penyiraman otomatis

Saat dijalankan, sensor kelembaban tanah mendeteksi kondisi tanah. Dari pengukuran yang didapat melalui sensor tersebut akan didapat 2 kondisi, yaitu tanah yang memiliki kelembaban kering dan tanah yang memiliki kelembaban basah. Jika sensor mendeteksi tanah kering kelembabannya dibawah 50 % maka LCD akan menampilkan nilai kelembaban serta kondisi tanah dan relay akan on sehingga pompa air otomatis akan menyala untuk menyiram tanaman. Jika sensor mendeteksi tanah basah kelembabannya diatas 50 % maka LCD akan menampilkan nilai kelembaban serta kondisi tanah dan relay akan off sehingga pompa air akan mati.



Gambar 41. ketika kondisi tanah basah



Gambar 42. Ketika kondisi tanah kering dan pompa menyiram

## Script

```
#include <LiquidCrystal.h>

#include <Wire.h> // i2C Connection Library
#include <LiquidCrystal_I2C.h> //Memanggil i2C
LCD Library

int pompa = 7;

LiquidCrystal_I2C lcd(0x27,16,2); // if your LCD
I2C address is not 0x27 , you should replace 0x27
with correct I2C address(i.e 0x3f)

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

 lcd.begin(16, 2);
 pinMode(3, INPUT);
 pinMode(13, INPUT);
 pinMode(7, OUTPUT);
 lcd.init(); // initialize the lcd
 lcd.backlight();
}

void loop() {
 int soilmos;

 soilmos=1023-analogRead(A0);//get soil moisture
value from A0 pin
 soilmos=map(soilmos, 0, 1023, 0, 100); //convert
moisture value in percentage format

 lcd.setCursor(0,0);
 lcd.print("Kelembaban = ");
 lcd.print(soilmos);
```

```

 lcd.print("%");
 Serial.print("Kelembaban = ");
 Serial.println(soilmos);

 if (soilmos<50){ digitalWrite(7,
 HIGH); lcd.setCursor(0, 1);
 lcd.print("Tanah Kering");
 delay(200);
 }

 else {
 digitalWrite(7, LOW);
 lcd.setCursor(0, 1); lcd.print("Tanah
 Basah "); delay(600);
 }
}

```

#### **D. Evaluasi**

1. Buatlah Project Sederhana dengan menggunakan minimal 2 Sensor dan 1 LCD

#### **E. Referensi**

1. Dian Artanto, (2017). *Interface Sensor dan Aktutor Menggunakan Proteus,Arduino dan Lab View*, Deepublish, Yogyakarta.
2. Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*.Apreest 1<sup>st</sup> Edition
3. Heri Andrianto dan Aan Darmawan (2016).*Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
4. Abdul Kadir (2018),*Arduino dan Sensor*, Andi Ofset, Yogyakarta

## Daftar Pustaka

- J. S. Wakur, *Alat Penyiram Tanaman Otomatis Menggunakan Arduino Uno*. 2015.
- L. . F. A. Caesar Pats Yahwe, Isnawaty, “Rancang Bangun Prototype System Monitoring Kelembaban Tanah Melalui Sms Berdasarkan Hasil Penyiraman Tanaman System Monitoring Kelembaban Tanah Melalui Sms Berdasarkan Hasil Penyiraman Tanaman,” *semanTIK*, vol. Vol 2, no. 1, pp. 97–110, 2016.
- H. W. Wijaya and B. S. Nugraha, “Perancangan alat penyiram tanaman otomatis dengan YL69 berbasis arduino uno r3 naskah publikasi,” 2017.
- T. Elektro, “Jurnal Teknologi Elektro , Universitas Mercu Buana RANCANG BANGUN SISTEM PENGAIRAN TANAMAN MENGGUNAKAN SENSOR KELEMBABAN TANAH,” vol. 8, no. 2, pp. 151–155, 2017.
- F. Amir, D. Rahmawati, and M. Ulum, “Penyiraman Tanaman Media Otomatis Berbasis Telepon Seluler PIntar dan Jaringan Sensor Fuzzy Tanpa Kabel,” *Semin. Nas. Mat. dan Apl. Univ. Airlangga.*, 2017
- C. R. Hidayat and F. D. Syahrani, “SAMPAH MENGGUNAKAN SENSOR PIR DAN SENSOR DESIGN OF CONTROL SYSTEM ON THE TRASH USING,” 1978.
- Agung, F. S., & Farhan, M. (2009). Sistem Deteksi Asap Rokok Pada Ruangan Bebas Asap Rokok Dengan Keluaran Suara, 1–9.
- Abdul Kadir (2017). *Pemrograman Arduino dan Prossesing*, PT.Elex Media Komputindo, Jakarta
- Harlod Timmis (2011). *Practical Arduino Engineering*. Apress 1<sup>st</sup> edition.
- Yuwono Martadinata., (2016). *Arduno itu Pintar* ,PT Elex Media Komputindo, Jakarta

- Hari Santoso (2015). *Panduan Praktis Arduino untuk Pemula*. Elang Sakti. Malang
- Desember, J., & Triyanto, A. (2015). Pengembangan Lampu LED Alternatif sebagai Efisiensi Daya, 7(2), 8–9.
- Dian Artanto, (2017). *Interface Sensor dan Aktutor Menggunakan Proteus, Arduino dan Lab View*, Deepublish, Yogyakarta.
- Rick Anderson and Dan Carvo ( 2013), *Pro Arduino*. Apreest 1<sup>st</sup> Edition
- Heri Andrianto dan Aan Darmawan (2016). *Arduino Belajar Cepat dan Pemrograman*. Informatika, Bandung.
- Abdul Kadir (2018), *Arduino dan Sensor*, Andi Offset, Yogyakarta

# MIKROKONTROLER 2

## Sekilas Tentang Arduino

Arduino adalah mikrokontroler yang sangat mudah penggunaanya dan menggunakan bahasa pemrograman C dengan IDE yang free dan bisa digunakan oleh siapa saja.

Dalam buku ini dibahas mengenai struktur bahasa yang digunakan, contoh penggunaan arduino dengan LDR, PIR, MQ-3, MQ-7, Sensor Suara, DHT 11, Ultrasonik, LCD 16x, dan disertai dengan contoh Project Arduino DIY yang dapat diterapkan di masyarakat sehingga mahasiswa dan masyarakat dapat mudah mempelajari dan mempraktekannya karena disertai contoh skript dan skematiknya

## Arduino dan Sensor

pada project Arduino DIY



Dani Sasmoko ST., M.Eng adalah pengajar di Universitas STEKOM yang berkecimpung dalam penelitian di bidang mikrokontroler khususnya ke bidang Internet Of Things . Lulusan S2 Elektro dari UGM dan S1 dari Universitas Islam Indonesia ( UII )

ISBN 978-623-6141-37-3 (PDF)

