

PROJECT SISTEM KENDALI ELEKTRONIK
BERBASIS
ARDUINO

Dr. Junaidi, S.Si., M.Sc
Yuliyani Dwi Prabowo

Perpustakaan Nasional RI:
Katalog Dalam Terbitan (KDT)

**Project Sistem Kendali Elektronik Berbasis
ARDUINO**

Penulis:

Dr. Junaidi, S.Si., M.Sc
Yuliyani Dwi Prabowo

Desain Cover & Layout

Team Aura Creative

Penerbit

AURA

CV. Anugrah Utama Raharja

Anggota IKAPI

No.003/LPU/2013

viii + 122 hal : 15,5 x 23 cm

Cetakan, Maret 2018

ISBN: 978-602-5636-46-2

Alamat

Jl. Prof. Dr. Soemantri Brojonegoro, Komplek Unila

Gedongmeneng Bandar Lampung

HP. 081281430268

E-mail : redaksiaura@gmail.com

Website : www.aura-publishing.com

Hak cipta pada penulis

Hak penerbitan pada penerbit

Tidak boleh diproduksi sebagian atau seluruhnya dalam bentuk apapun

Tanpa izin tertulis dari pengarang dan/atau penerbit

Kutipan Pasal 72 :

Sanksi pelanggaran Undang-undang Hak Cipta (UU No. 10 Tahun 2012)

1. Barang siapa dengan sengaja dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 ayat (1) atau Pasal (49) ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp. 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan atau denda paling banyak Rp. 5.000.000.000,00 (lima miliar rupiah)
2. Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu Ciptaan atau hasil barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah)

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan karunia-NYA, sehingga kami dapat menyelesaikan buku “Project Sistem Kendali Elektronik Berbasis ARDUINO” untuk Mahasiswa dan Umum. Arduino adalah platform prototyping open-source hardware yang dapat digunakan untuk membuat proyek berbasis pemrograman. Arduino dirancang untuk memudahkan pengguna elektronik atau siapapun yang ingin mengembangkan peralatan elektronik interaktif berdasarkan hardware dan software yang fleksibel dan mudah untuk digunakan.

Buku ini menjelaskan tentang Mikrokontroler dan Arduino secara lengkap dan disertai dengan contoh-contoh Project berbasis Arduino. Buku ini berisikan tentang Dasar-dasar mikrokontroler, Pengenalan Arduino, Pengenalan Aplikasi Arduino IDE, dan berbagai project berbasis Arduino.

Kami ucapkan terima kasih kepada semua pihak yang turut membantu dalam penulisan dan penerbitan buku ini. Kami sadari, masih banyak kekurangan dan kelemahan dalam penyusunan buku ini, untuk itu kritik dan saran akan penulis terima dengan senang hati. Semoga buku ini dapat memberikan manfaat bagi kita semua.

Bandar Lampung, 14 Maret 2018

Penulis

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	iv
BAB 1 Pengenalan Arduino	1
<hr/>	
1.1 Dasar-Dasar Mikrokontroler dan Arduino	1
1.2 Jenis-Jenis Arduino	4
1.2.1 Arduino Uno	5
1.2.2 Arduino Due	10
1.2.3 Arduino Mega	12
1.2.4 Arduino Leonardo	15
1.2.5 Arduino Fio	19
1.2.6 Arduino Lilypad	21
1.2.7 Arduino Nano	22
1.2.8 Arduino Pro Mini	23
1.2.9 Arduino Micro	25
BAB 2 Pengenalan Aplikasi Arduino IDE	26
<hr/>	
2.1 Instalasi Arduino IDE	27
2.2 Penggunaan Aplikasi Arduino IDE	33

BAB 3	Project 1 - LED Berkedip	40
	3.1 Light Emitting Diode (LED)	42
	3.2 Rangkaian Lampu Berkedip	43
	3.3 Program Lampu Berkedip	44
	3.4 Upload Program ke Board Arduino Uno	48
BAB 4	Project 2 - LED Lampu Lalu Lintas	51
	4.1 Rangkaian Simulasi Lampu Lalu Lintas	52
	4.2 Program Simulasi Lampu Lalu Lintas	53
BAB 5	Project 3 - Sensor Cahaya (LDR)	56
	5.1 Rangkaian Sensor Cahaya	58
	5.2 Program Sensor Cahya	59
BAB 6	Project 4 - Sensor Gerak (PIR)	62
	6.1 Rangkaian Sensor Gerak	63
	6.2 Program Sensor Gerak	65
BAB 7	Project 5 - Module LCD 16x2 dengan I2C Konverter	68
	7.1 Modul LCD 16x2 dengan I2C Konverter	68
	7.2 Menambahkan Library IDE Arduino	70
	7.3 Rangkaian LCD 16x2 dengan I2C Konverter	75
	7.4 Program	76

BAB 8	Project 6 - Sensor Jarak (HC-SR04) dengan Serial Monitor	79
	8.1 Sensor Ultrasonik	79
	8.2 Rangkaian Sensor Jarak	82
	8.3 Program Sensor Jarak	84
BAB 9	Project 7 - Sensor Jarak (HC-SR04) dengan Modul LCD 16x2	86
	9.1 Rangkaian	86
	9.2 Program	88
BAB 10	Project 8 - Sensor Suhu (LM35)	92
	10.1 Sensor Suhu LM35	92
	10.2 Rangkaian Sensor Suhu	94
	10.3 Program Sensor Suhu	95
BAB 11	Project 9 - Sensor Suhu (LM35) dengan Modul LCD 16x2	97
	11.1 Rangkaian	97
	11.2 Program	99
BAB 12	Project 10 - Kontrol Motor Stepper	102
	12.1 Motor Stepper	102
	12.2 Rangkaian	105
	12.3 Program	106

BAB 13	Project 11 - Kontrol Motor DC	108
<hr/>		
13.1	Rangkaian	109
13.1	Program	109
BAB 14	Project 12 - Alarm Kebakaran (IR Flame)	111
<hr/>		
14.1	Sensor Api Infrared	111
14.2	Rangkaian	112
14.3	Program	113
DAFTAR PUSTAKA		116
BIOGRAFI PENULIS		117
<hr/>		

Pengenalan Arduino

1.1 Dasar-Dasar Mikrokontroler dan Arduino

Sebelum membahas lebih detail tentang Arduino, marilah kita sama-sama pahami tentang dasar-dasar mikrokontroler. Menurut pengertian atau istilah yang sering digunakan, mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik yang terdiri dari CPU (*Central Processing Unit*), memori, I/O (*input/output*), bahkan sudah dilengkapi dengan ADC (*Analog-to-Digital Converter*) yang sudah terintegrasi di dalamnya. Kelebihan utama dari mikrokontroler adalah tersedianya RAM (*Random Access Memory*) dan peralatan I/O pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas.

Mikrokontroler pertama kali dikenalkan oleh Texas Instrument pada tahun 1974 dengan seri TMS-1000. Mikrokontroler pertama ini merupakan mikrokontroler 4 bit. Mikrokontroler ini memiliki sebuah chip yang telah dilengkapi dengan RAM dan ROM (*Read Only Memory*). Selanjutnya, pada tahun 1976 Intel mengeluarkan mikrokontroler 8 bit dengan nama 8748 yang merupakan mikrokontroler dari keluarga MCS-48. Untuk saat ini telah pasaran

banyak ditemui tipe-tipe mikrokontroler mulai dari 8 bit sampai dengan 64 bit. Masing-masing fabrikaan (vendor) mengeluarkan mikrokontroler yang telah dilengkapi dengan berbagai fasilitas sehingga memudahkan pengguna (*user*) untuk merancang sebuah sistem dengan komponen luar yang relatif lebih sedikit.

Saat ini mikrokontroler yang banyak beredar dipasaran adalah mikrokontroler 8 bit varian keluarga MCS51 dengan tipe CISC, yaitu *Complex Instruction Set Computing* yang dikeluarkan oleh Atmel dengan seri AT89Sxx, dan mikrokontroler AVR yang merupakan mikrokontroler RISC (*Reduced Instruction Set Computer*) dengan seri ATMegaxx. Dengan mikrokontroler tersebut pengguna sudah bisa membuat sebuah sistem untuk keperluan sehari-hari, seperti pengendali peralatan rumah tangga jarak jauh yang menggunakan remote control televisi (*smart home*), jam digital, text berjalan, termometer digital, dan lain-lain.

Dalam perkembangannya, modul atau minimum sistem dari mikrokontroler dibuat dalam bentuk chip yang lebih memudahkan pengguna untuk menggunakannya. Satu hal yang saat ini sedang atau banyak digemari oleh pengguna mikrokontroler adalah modul Arduino.

Ketika kita membahas tentang Arduino, maka timbul sebuah pertanyaan tentang apakah Arduino sama dengan mikrokontroler?. Mungkin diantara kita masih banyak yang akan mengatakan, “ya”. Akan tetapi, apabila kita merujuk pada pemahaman yang benar, maka Arduino tidak tepat jika dikatakan sebagai mikrokontroler. Secara detail, perbedaan dari keduanya dapat kita peroleh dari situs resmi Arduino (www.arduino.cc) sebagai berikut:

“Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments”

Apabila kita coba memahami makna dari pengertian di atas, maka Arduino dapat dikatakan sebagai ***prototyping platform***. Dalam Bahasa Indonesia, *prototype* dapat diartikan sebagai purwarupa, yaitu suatu alat yang dapat digunakan untuk menghasilkan berbagai karya cipta dalam tahapan desain. Pada kenyataannya, Arduino tidak hanya digunakan pada tahapan desain, namun sampai produk jadi. Kita dapat berkreasi apapun dengan menggunakan Arduino, seperti aplikasi dalam bidang robotika, atau aplikasi-aplikasi *embedded system* lainnya. Arduino memberikan banyak kemudahan bagi pengguna untuk merealisasikan karya-karyanya. Arduino telah dilengkapi dengan sistem IDE (*Intergrated Development Environment*) untuk menuliskan program aplikasi yang kita buat. Selain itu, pada board Arduino juga telah dilengkapi dengan berbagai fasilitas sehingga lebih memudahkan para pencinta atau penggunaanya.

Dari pemaparan di atas, maka dapat kita perjelas bahwa Arduino adalah *platform prototyping open-source hardware* yang dapat digunakan untuk membuat proyek berbasis pemrograman. *Hardware* Arduino memiliki prosesor mikrokontroler ATMega yang dirilis oleh Atmel AVR, tetapi *software* yang digunakan memiliki bahasa pemrograman tersendiri. Arduino dirancang untuk memudahkan pengguna elektronik atau siapapun yang ingin

mengembangkan peralatan elektronik interaktif berdasarkan *hardware* dan *software* yang fleksibel dan mudah untuk digunakan.

Sampai di sini, kita sudah harus memahami perbedaan antara mikrokontroler dengan Arduino. Mari kita sama-sama simpulkan, Arduino adalah papan elektronik yang menggunakan mikrokontroler jenis tertentu. Sebagai contoh adalah jenis Arduino yang paling banyak digunakan, yaitu Arduino UNO. Jenis ini menggunakan mikrokontroler keluaran Atmel, yaitu seri ATmega328.

Dalam berbagai aplikasi, Arduino dapat digunakan untuk mendeteksi lingkungan dengan menerima *input* dari berbagai sensor atau tombol (sensor cahaya, suhu, inframerah, ultrasonik, jarak, tekanan, kelembaban) dan dapat mengontrol perangkat lainnya seperti mengontrol kecepatan dan arah putar motor, menyalakan LED, dan sebagainya. Keuntungan yang kita dapatkan ketika menggunakan Arduino, antara lain:

- Harga relatif murah dibandingkan dengan mikrokontroler lainnya dengan kelebihan yang ditawarkan.
- Dapat digunakan pada berbagai sistem operasi Windows, Linux, Max, dan lain-lain.
- Memiliki bahasa pemrograman yang mudah dipahami, proyek Arduino sudah banyak dipelajari karena *open source*.

1.2 Jenis-Jenis Arduino

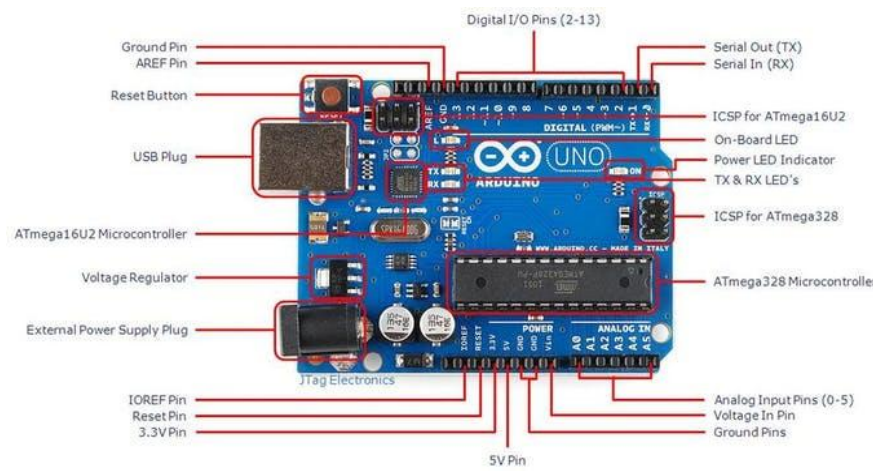
Banyak jenis Arduino mulai dari yang paling murah, mudah dicari, dan yang paling banyak digunakan seperti Arduino Uno atau Nano,

hingga Arduino yang berbentuk mini PC. Karena sifatnya yang *open source*, maka banyak sekali vendor yang membuat dan menjual produk-produk berbasis Arduino. Berikut ini adalah jenis-jenis Arduino yang banyak beredar dipasaran.

1.2.1 Arduino Uno

Arduino Uno merupakan salah satu Arduino yang murah, mudah didapat, dan sering digunakan. Arduino Uno ini dibekali dengan mikrokontroler ATMEGA328P dan versi terakhir yang dibuat adalah versi R3. Modul ini sudah dilengkapi dengan berbagai hal yang dibutuhkan untuk mendukung mikrokontroler untuk bekerja.

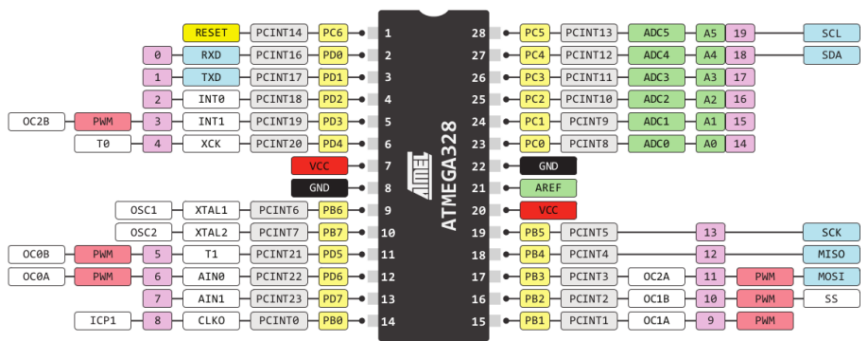
ATMega328P yang sudah terbentuk modul Arduino uno seperti terlihat pada gambar di bawah ini.



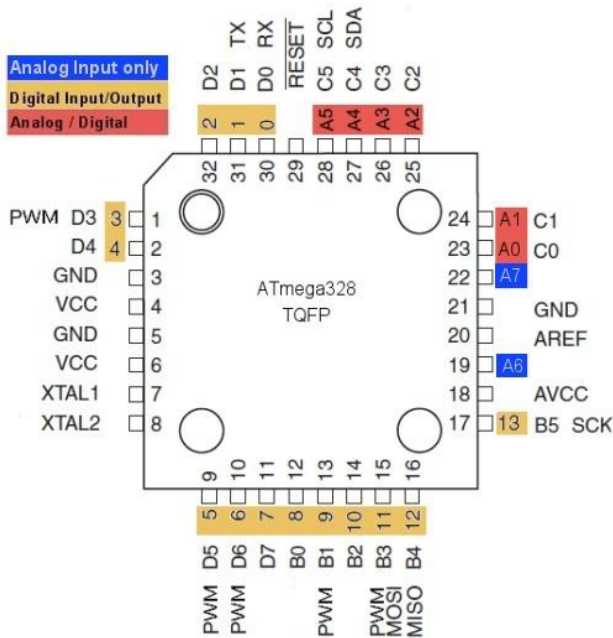
Gambar 1.1 Arduino Uno R3

ATMega328 adalah mikrokontroller keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*)

dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*).



(a)



(b)

Gambar 1.2 (a) Pinout ATmega328 model DIP dan (b) Pinout ATmega328 model SMD

ATmega328 memiliki beberapa fitur antara lain:

- a. Memiliki 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus clock.

- b. Memiliki 32 x 8-bit register serba guna.
- c. Kecepatan akses mencapai 16 MIPS dengan clock 16 MHz.
- d. Memiliki 32 KB Flash memory dan pada Arduino memiliki bootloader yang menggunakan 2 KB dari flash memori sebagai bootloader.
- e. Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.
- f. Memiliki SRAM (*Static Random Access Memory*) sebesar 2 KB.
- g. Memiliki pin I/O digital sebanyak 14 pin, 6 pin diantaranya dapat digunakan sebagai pin PWM (*Pulse Width Modulation*).
- h. Memiliki Master/Slave SPI Serial interface.

Mikrokontroler ATmega328 memiliki arsitektur Harvard, yaitu pemisah antara memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dari mikrokontroler. Instruksi – instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program.

Konsep inilah yang memungkinkan instruksi – instruksi dapat dieksekusi dalam setiap satu siklus clock. Sebanyak 32 x 8-bit register serba guna digunakan untuk mendukung operasi pada ALU (*Arithmetic Logic Unit*) yang dapat dilakukan dalam satu siklus. Sebanyak 6 dari register serbaguna ini dapat digunakan sebagai 3 buah register pointer 16-bit pada mode pengalamatan tidak

langsung untuk mengambil data pada ruang memori data. Ketiga register pointer 16-bit ini disebut dengan register X (gabungan R26 dan R27), register Y (gabungan R28 dan R29), dan register Z (gabungan R30 dan R31). Hampir semua instruksi AVR memiliki format 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit.

Selain register serba guna di atas, terdapat register lain yang terpetakan dengan teknik *memory mapped I/O* berukuran 64 byte. Beberapa register ini digunakan untuk fungsi khusus antara lain sebagai register control Timer/Counter, Interupsi, ADC, USART, SPI, EEPROM, dan fungsi I/O lainnya. Register – register ini menempati memori pada alamat 0x20h – 0x5Fh.

Arduino Uno R3 memiliki 14 pin digital I/O (dimana 6 pin dapat digunakan sebagai *Output PWM*), 6 pin analog *input*, 2x3 pin ICSP (untuk memprogram Arduino dengan software lain), dan kabel USB. Untuk menghidupkannya cukup dengan menghubungkan kabel USB ke komputer atau menggunakan adaptor 5 VDC. Arduino ini sangat disarankan untuk untuk pemula yang ingin belajar Arduino.

Berikut ini adalah spesifikasi untuk Arduino Uno R3 yang dikutip dari <https://store.Arduino.cc/usa/Arduino-uno-rev3>.

Parameter	Spesifikasi
Tegangan Operasi	5V
Tegangan Input (Rekomendasi)	7-12V
Tegangan Input (Limit)	6-20V
Pin Digital I/O	14 (6 PinOutput PWM)

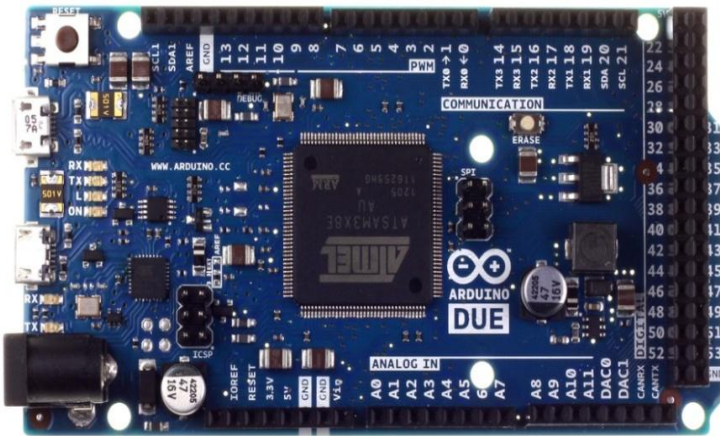
Pin Digital PWM I/O	6
Pin Analog Input	6
Arus DC tiap Pin I/O	20 mA
Arus DC Pin 3.3V	50 mA
Flash Memory	32 KB (ATmega328P) dimana 0.5 KB digunakan untuk bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Panjang	68,6 mm
Lebar	53,4 mm
Berat	25 g

Arduino Uno memiliki kelebihan-kelebihan yang membuat tipe Arduino ini menjadi lebih mudah dan menyenangkan, antara lain:

- Pengembangan *project* mikrokontroler akan menjadi lebih mudah dan menyenangkan. Pengguna dapat langsung menghubungkan board Arduino ke komputer atau laptop melalui kabel USB. Board Arduino juga tidak membutuhkan downloader untuk mendownloadkan program yang telah dibuat dari computer ke mikrokontrolernya.
- Didukung oleh Arduino IDE dengan bahasa pemrograman dengan library yang lengkap.
- Terdapat modul yang siap pakai/shield sehingga dapat langsung dipasang pada board Arduino.

1.2.2 Arduino Due

Arduino Due adalah pengembangan dari mikrokontroler Arduino yang menggunakan CPU Atmel SAM3X8E ARM Cortex-M3. Arduino Due adalah Arduino yang pertama kali menggunakan prosesor ARM 32-bit Cortex-M3 keluaran Atmel. Dengan demikian, Arduino Due adalah Arduino Development Board pertama yang dibuat dengan menggunakan mikrokontroler ARM 32-bit.



Gambar 1.3 Arduino Due

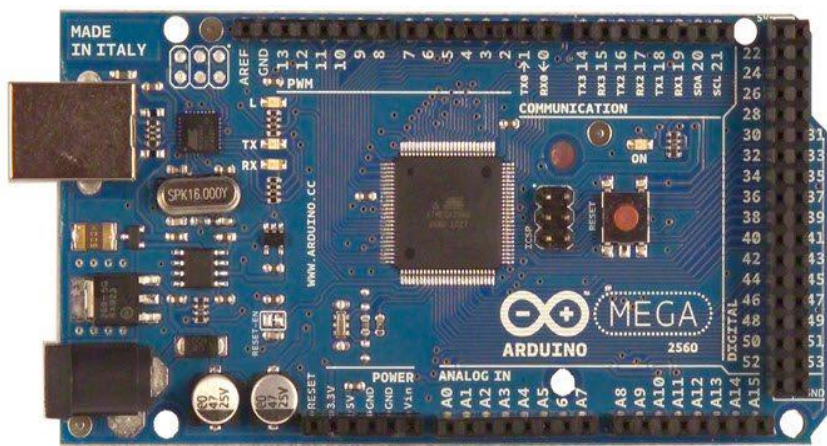
Arduino ini memiliki 54 pin I/O digital (dimana 12 pin dapat digunakan sebagai *output* PWM), 12 pin analog *Input*, 4 pin UART (pin serial untuk komunikasi serial asinkron TX/RX), dan dilengkapi dengan Micro USB yang umumnya terdapat pada handphone saat ini.

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

- a. Memiliki core processor 32-bit yang memungkinkan operasi data sebanyak 4 byte sekaligus dengan tipe data DWORD pada satu siklus waktu.
- b. CPU dilengkapi dengan frekuensi clock sebesar 84 MHz.
- c. RAM statis (SRAM) sebesar 96 KB (48 kali lipat lebih besar dibanding Arduino Uno, 12 kali lipat lebih besar dibanding Arduino Mega 2560).
- d. 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).
- e. Terdapat pengendali akses memori langsung (*Direct Memory Access* atau DMA controller) yang dapat membebaskan CPU dari operasi memori yang intensif.
- f. 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).
- g. Resolusi ADC yang lebih presisi dengan 12-bit ($2^{12} = 4096$ jenjang, 0-4095) sebanyak 12 kanal.
- h. Sumber catu daya dapat diambil dari port USB atau catu daya eksternal antara 6 ~ 16 Volt DC (direkomendasikan antara 7 ~ 12 VDC).

1.2.3 Arduino Mega

Arduino Mega umumnya dibuat menggunakan jenis mikrokontroler ATmega 2560. Sesuai dengan namanya, Arduino ini dibekali dengan prosesor ATmega2560 yang memiliki 54 pin digital I/O (dimana 15 pin dapat digunakan sebagai output PWM), 16 pin analog input, 4 pin UART, 2x3 pin ICSP (untuk memprogram Arduino dengan software lain), dan kabel USB komputer yang sekaligus digunakan sebagai sumber tegangan.



Gambar 1.4 Arduino Mega 2560

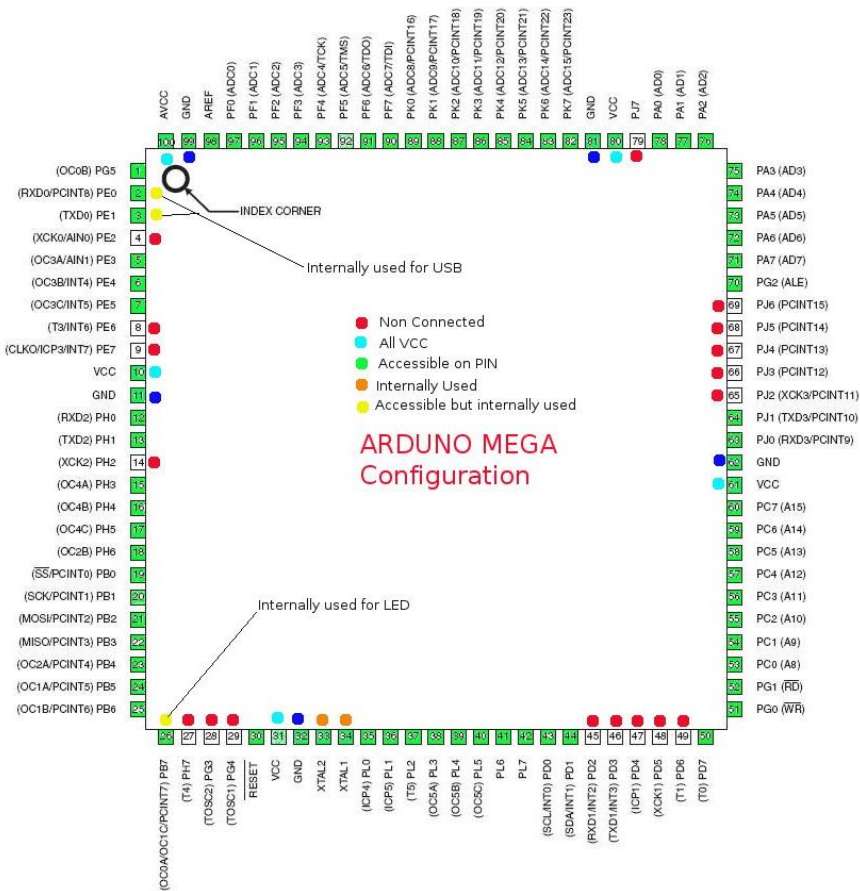
Spesifikasi dari Arduino mega 2560 secara lebih lengkap seperti diuraikan sebagai berikut:

Parameter	Spesifikasi
Tegangan Operasional	5V
Tegangan Input (rekomendasi)	7-12V
Tegangan Input (limit)	6-20V
Pin Digital I/O	54 (15 PWM)
Pin Analog Input	16
Arus DC per Pin I/O	20 mA

Arus DC untuk Pin 3.3 V	50 mA
Memori Flash	256 KB (8 KB: bootloader)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Panjang	101,52 mm
Lebar	53,3 mm
Berat	37 g

Arduino Mega 2560 dilengkapi dengan 54 Pin digital yang dapat digunakan sebagai input atau output dan 16 Pin Analog berlabel A0 sampai A15 sebagai ADC. Setiap Pin Analog memiliki resolusi sebesar 10 bit. Arduino Mega 2560 juga dilengkapi dengan fitur yang memiliki fungsi khusus, sebagai berikut:

- a. Memiliki 4 buah masukan serial, yaitu Port Serial 0: Pin 0 (RX) dan Pin 1 (TX); Port Serial 1: Pin 19 (RX) dan Pin 18 (TX); Port Serial 2: Pin 17 (RX) dan Pin 16 (TX); Port Serial 3: Pin 15 (RX) dan Pin 14 (TX). Pin Rx digunakan untuk menerima data serial TTL dan Pin (Tx) untuk mengirim data serial TTL.
- b. Memiliki external Interrupts sebanyak 6 buah: Pin 2 (Interrupt 0), Pin 3 (Interrupt 1), Pin 18 (Interrupt 5), Pin 19 (Interrupt 4), Pin 20 (Interrupt 3), dan Pin 21 (Interrupt 2).



Gambar 1.5 Pinout ATmega2560

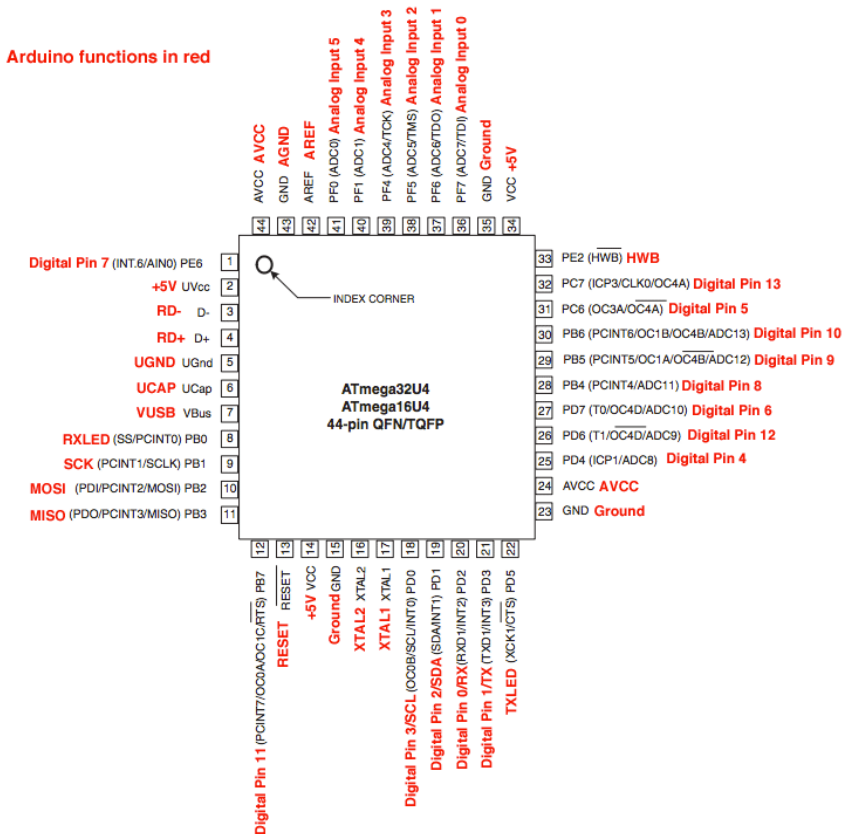
- Memiliki 15 buah PWM, yaitu pada pin: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, dan 44, 45, dan 46. Pin-pin tersebut dapat digunakan sebagai output PWM berukuran 8 bit.
- Pin I2C: Pin 20 (SDA) dan Pin 21 (SCL), Komunikasi I2C menggunakan wire library.
- Pin SPI: Pin 50 (MISO), Pin 51 (MOSI), Pin 52 (SCK), Pin 53 (SS), digunakan untuk komunikasi SPI menggunakan SPI Library.

1.2.4 Arduino Leonardo

Arduino Leonardo adalah jenis Arduino yang dibuat menggunakan mikrokontroler ATmega32u4. Arduino Leonardo memiliki 20 pin digital I/O (dimana 7 pin dapat digunakan sebagai Output PWM), 12 pin analog Input, 2x3 pin ICSP, dan juga dibekali dengan Mikro USB seperti Arduino Due. Dilihat dari bentuk dan spesifikasinya, Arduino Leonardo memiliki kemiripan dengan Arduino Uno. Perbedaannya hanya terletak pada penggunaan micro USB untuk pemrograman sekaligus sumber tegangan.

Arduino Leonardo berbeda dari semua papan Arduino yang lainnya karena ATmega32u4 secara terintegrasi (*built-in*) telah memiliki komunikasi USB, sehingga tidak lagi membutuhkan prosesor sekunder (tanpa chip ATmega16U2 sebagai konverter USB-to-serial). Hal ini memungkinkan Arduino Leonardo yang terhubung ke komputer digunakan sebagai mouse dan keyboard, selain bisa digunakan sebagai virtual (CDC) serial/COM port.

Mikrokontroler ATmega32u4 merupakan microchip 8-bit AVR RISC berkapasitas rendah yang menampilkan memori program flash berkapasitas 32 KB (self-programming), memiliki SRAM 2,5 KB, 1 KB EEPROM, perangkat kecepatan berkecepatan tinggi USB 2.0/kecepatan rendah, 12-channel 10-bit ADC, dan antarmuka JTAG. Perangkat ini mencapai hingga 16 throughput MIPS pada 16 MHz dan tegangan operasi sekitar 2,7 – 5,5 volt. Pinout dari ATmega32u4 seperti tampak pada **Gambar 1.6**.



Gambar 1.6 Pinout ATmega32u4.



Gambar 1.7 Arduino Leonardo

Pin tegangan yang tersedia pada papan Arduino Leonardo adalah sebagai berikut:

- a. Tegangan V_{IN} : Adalah input tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai ‘saingan’ tegangan 5 volt dari koneksi USB atau sumber daya ter-regulator lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui jack power, kita bisa mengakses/mengambil tegangan melalui pin ini.
- b. Tegangan 5V: Tegangan listrik ter-regulator yang digunakan untuk daya mikrokontroler dan komponen lainnya pada papan Arduino. Tegangan dapat menggunakan pin VIN melalui regulator on-board, atau dipasok oleh USB atau power suplai lain dengan besar tegangan 5V ter-regulator.
- c. Tegangan 3V3: Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (on-board). Arus maksimum yang dihasilkan adalah 50 mA.
- d. Pin GND: Pin Ground atau Massa.
- e. Pin IOREF: Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler (atau VCC untuk papan). Pin ini bertegangan 5V pada Leonardo.

20 pin digital I/O pada Leonardo dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Mereka beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal sebesar 20-50 kOhm yang terputus secara default. Selain itu, beberapa pin memiliki fungsi khusus, yaitu:

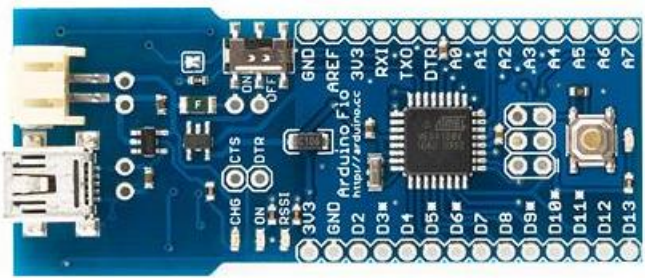
- a. Serial: Pin 0 (RX) dan pin 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL menggunakan hardware ATmega32U4 yang memiliki kemampuan serial didalamnya. Perhatikan bahwa pada Leonardo, kelas Serial mengacu pada komunikasi USB (CDC); untuk TTL serial pada pin 0 dan 1, menggunakan kelas Serial 1.
- b. TWI: Pin 2 (SDA) dan pin 3 (SCL). Dukungan komunikasi TWI menggunakan sistem Wire.
- c. Eksternal Interupsi: Pin 3 (interrupt 0), pin 2 (interrupt 1), pin 0 (interrupt 2), pin 1 (interrupt 3) dan pin 7 (interrupt 4). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau merubah nilai.
- d. Pin PWM: Pin 3, 5, 6, 9, 10, 11, dan 13 berukuran 8-bit dengan fungsi analogWrite().
- e. Pin SPI: Pin pada header ICSP ini mendukung komunikasi SPI menggunakan sistem SPI. Pada tipe Arduino ini pin SPI tidak terhubung ke satu pun pin digital I/O karena yang terhubung langsung hanya pada Arduino Uno yang hanya menyediakan konektor ICSP. Ini berarti bahwa jika kita memiliki shield yang menggunakan SPI, tetapi tidak terdapat 6 pin konektor ICSP yang terhubung ke 6 pin ICSP header Arduino Leonardo, maka shield tidak akan bekerja.
- f. LED: Pin 13. Tersedia secara built-in pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diset bernilai LOW, maka LED padam (OFF).

- g. Input Analog: Pin A0-A5, Pin A6 - A11 (pada pin digital 4, 6, 8, 9, 10, dan 12). Arduino Leonardo memiliki 12 input analog, berlabel A0 sampai A11, yang semuanya juga dapat digunakan sebagai digital I/O. Pin A0-A5 terdapat di lokasi yang sama seperti pada Arduino Uno; Pin input A6-A11 masing-masing ada pada digital I/O pin 4, 6, 8, 9, 10, dan 12. Masing-masing pin menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai 0-5 volt, dan memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah dengan menggunakan pin AREF dan fungsi `analogReference()`.
- h. AREF: Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.
- i. RESET: Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino.

1.2.5 Arduino Fio

Arduino Fio merupakan Arduino yang memiliki bentuk unik yang dibekali dengan prosesor ATmega328V yang bekerja pada tegangan 3,3V dan frekuensi 8 MHz. Arduino Fio memiliki 14 pin digital I/O (6 pin bisa digunakan untuk *output* PWM), 8 pin analog input, dan memiliki socket USB to Xbee. Pengguna dapat meng-upload program dengan kabel FTDI.

Selain itu, dengan menggunakan modifikasi USB – to- XBee adaptor seperti XBee Explorer USB dengan socket USB to Xbee, memungkinkan pengguna dapat menggunggah program secara nirkabel.



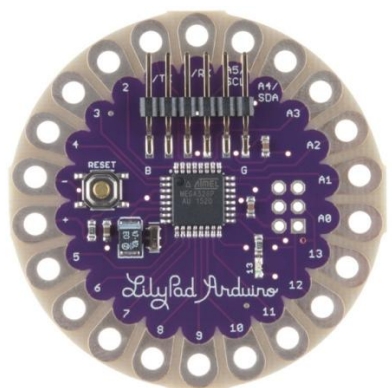
Gambar 1.8 Arduino Fio

Spesifikasi dari Arduino Fio adalah sebagai berikut:

Parameter	Spesifikasi
Tegangan Operasi	3,3V
Tegangan input	3,35 -12 V
Tegangan input untuk charge	3.7 - 7 V
Pin Digital I/O	14 (6 buah PWM)
Pin Analog Input	8
Pin DC per I/O	40 mA
Memory Flash	32 KB (2 KB bootloader)
SRAM	2 KB
EEPROM	1 KB
Kecepatan clock	8 MHz
Lebar	28 mm
Panjang	65 mm
Berat	9 g

1.2.6 Arduino Lilypad

Arduino Lilypad merupakan jenis Arduino yang sangat unik karena memiliki bentuk yang berbeda dari kebanyakan Arduino. Arduino Lilypad berbentuk melingkar seperti ditunjukkan pada gambar berikut.



Gambar 1.9 Arduino Lilypad

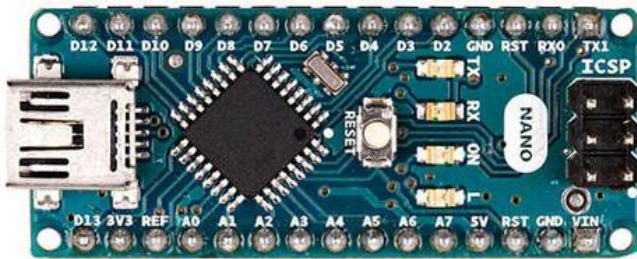
Arduino tipe ini dibekali dengan prosesor ATmega168V (versi daya rendah) dan ATmega328V yang sering digunakan untuk membuat proyek-proyek unik. Arduino ini dilengkapi dengan 14 pin digital I/O, dan 6 pin input analog. Spesifikasi dari Arduino Lilypad seperti dijelaskan sebagai berikut:

Parameter	Spesifikasi
Tegangan Operasi	2,7-5,5 V
Tegangan input	2,7-5,5 V
Pin Digital I/O	14 (6 buah PWM)
Pin Analog Input	6
Arus DC per Pin I/O	40 mA
Memory Flash	16 KB (2 KB bootloader)
SRAM	1 KB

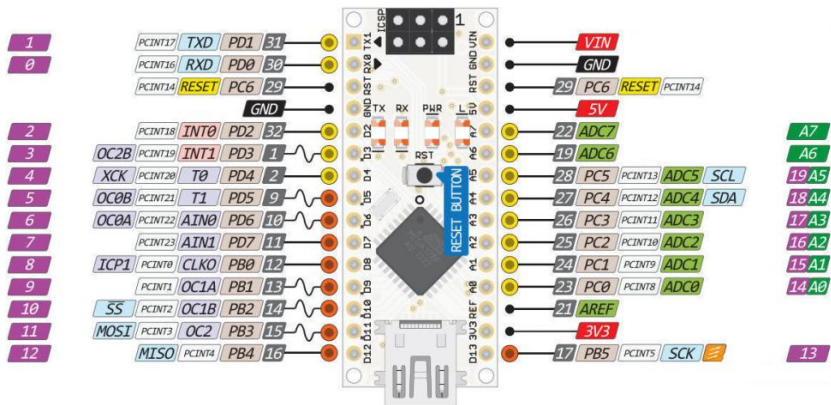
EEPROM	512 B
Kecepatan clock	8 MHz

1.2.7 Arduino Nano

Sesuai dengan namanya, Arduino Nano memiliki ukuran yang relatif kecil dan sangat sederhana. Dengan ukurannya yang kecil, bukan berarti jenis Arduino ini tidak mampu menyimpan banyak fasilitas. Hampir sama dengan tipe Arduino UNO, Arduino Nano dibekali dengan prosesor ATmega328P dengan bentuk SMD dan memiliki 14 Pin Digital I/O, 8 Pin Analog Input (lebih banyak dari Uno), dan menggunakan FTDI untuk pemrograman lewat Mikro USB. Selain itu juga ada yang menggunakan prosesor ATmega168.



Gambar 1.10 Arduino Nano



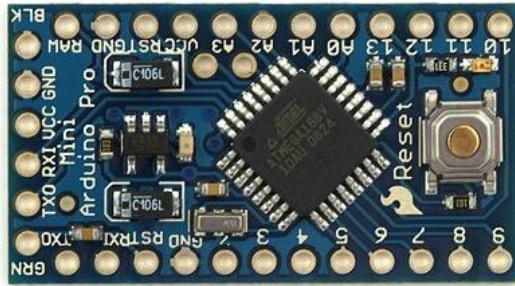
Gambar 1.11 Pinout Arduino Nano

Spesifikasi dari Arduino nano adalah sebagai berikut:

Parameter	Spesifikasi
Tegangan operasi	5V
Tegangan input (rekomendasi)	7V - 12V
Digital I/O pin	14 buah, 6 diantaranya menyediakan PWM
Pin Analog Input	8 buah
Arus DC per pin I/O	40 mA
Memori Flash	32 KB, 0,5 KB telah digunakan untuk bootloader
SRAM	2 KB
EEPROM	1 KB
Clock speed	16 Mhz
Dimensi	45 mm x 18 mm
Berat	5 g

1.2.8 Arduino Pro Mini

Arduino Pro Mini merupakan jenis Arduino dengan spesifikasi sama dengan yang dimiliki Arduino Nano. Perbedaannya terdapat pada tidak adanya fitur Micro USB untuk pemrograman. Untuk memasukan program ke Arduino jenis ini, diperlukan downloader khusus untuk memasukkan program kedalamnya. Arduino ini yang dibekali prosesor ATmega328 yang memiliki 14 pin digital I/O (6 pin dapat digunakan sebagai *output* PWM), 6 pin analog *Input*. Arduino ini cocok untuk proyek yang bersifat permanen di dalam suatu alat.



Gambar 1.12 Arduino Pro Mini

Spesifikasi dari Arduino pro mini adalah sebagai berikut:

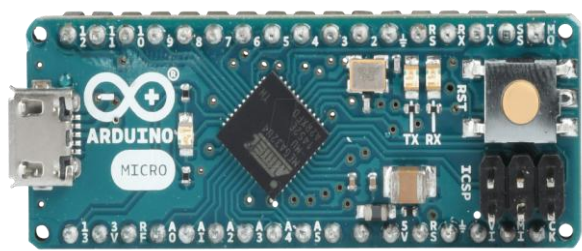
Parameter	Spesifikasi
Tegangan operasi	5V atau 3,3V (tergantung model)
Digital I/O pin	14 buah, 6 diantaranya menyediakan PWM
Analog Input pin	6 buah
Arus DC per pin I/O	40 mA
Memori Flash	32 KB; 0,5 KB telah digunakan untuk bootloader
SRAM	2 KB
EEPROM	1 KB
Clock speed	8 Mhz (model 3,3V) atau 16 Mhz (Model 5V)
Dimensi	33 mm x 18 mm
Berat	5 g

Untuk memasukkan program ke mikrokontroler, Arduino Pro Mini menggunakan tambahan alat berupa CH340G Converter yang digunakan sebagai downloader. Berikut ini adalah konfigurasi dari pin Arduino Pro Mini dengan model CH340G Converter.

CH340G Converter		Arduino Pro Mini
DTR	→	DTR
RXD	→	TXO
TXD	→	RXI
VC	→	VCC
CTS	→	<i>Not Connected</i>
GND	→	GND

1.2.9 Arduino Micro

Arduino Micro adalah suatu jebis Arduino yang dibekali dengan mikrokontroler ATmega32U4. Ukurannya lebih panjang dari Arduino Nano dan Arduino Pro Mini, karena memang fasilitasnya lebih banyak yaitu dengan 20 pin digital I/O (7pin bisa digunakan untuk *Output* PWM), 12 pin analog *Input*, dan Mikro USB.



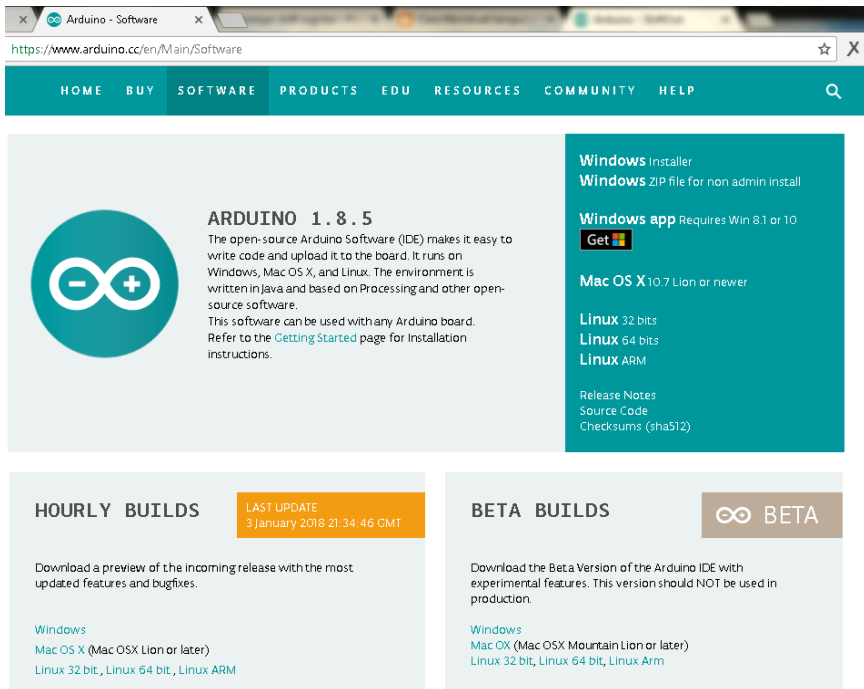
Gambar 1.13 Arduino Micro

Komunikasi serial pada Arduino Micro dilakukan dengan menggunakan Pin 0 (RX) dan pin 1 (TX) untuk menerima (RX) dan mengirimkan (TX) data serial TTL menggunakan hardware ATmega32U4. Perhatikan bahwa pada Arduino Micro, pengiriman data Serial mengacu pada komunikasi USB (CDC); untuk TTL serial pada pin 0 dan 1, menggunakan kelas Serial 1.

Pengenalan Aplikasi Arduino IDE

Pada bahasan kali ini akan dijelaskan tentang aplikasi Arduino IDE (*Integrated Development Environment*). Aplikasi Arduino IDE berfungsi untuk membuat, membuka, dan mengedit program yang akan kita masukkan ke dalam board Arduino. Aplikasi Arduino IDE dirancang agar memudahkan penggunaanya dalam membuat berbagai aplikasi. Arduino IDE memiliki struktur bahasa pemrograman yang sederhana dan fungsi yang lengkap, sehingga mudah untuk dipelajari oleh pemula sekalipun. Untuk mendapatkan aplikasi Arduino IDE dapat diunduh di Website resmi Arduino yaitu: <https://www.Arduino.cc/en/Main/Software>.

Website ini **Gambar 2.1**, menyediakan aplikasi Arduino IDE untuk beberapa sistem operasi komputer diantaranya Windows Installer/Non Installer, Mac OS, Linux 32 bits, Linux 54 bits, dan Linux ARM. Cukup dengan menekan link unduhnya (dalam hal ini kita pilih **Windows Installer**) maka akan muncul pilihan **download and donate** dan **just download**. Pilih just download maka secara otomatis file akan terunduh.



Gambar 2.1 Website resmi Arduino.cc
(<https://www.Arduino.cc/en/Main/Software>).

2.1 Instalasi Arduino IDE

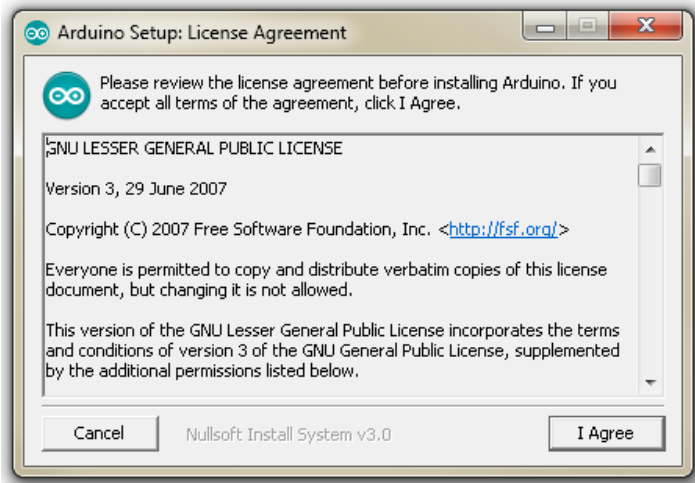
Pada bahasan ini akan dijelaskan cara instalasi aplikasi Arduino IDE pada sistem operasi Windows 7. Perlu diketahui, ada aplikasi Arduino IDE yang memerlukan instalasi dan ada yang tidak perlu instalasi.

Aplikasi Arduino IDE Installer

Untuk proses instalasi, buka file **Arduino-1.8.5-windows.exe** (1.8.5 adalah versi dari Arduino IDE), maka akan muncul seperti pada **Gambar 2.3**.

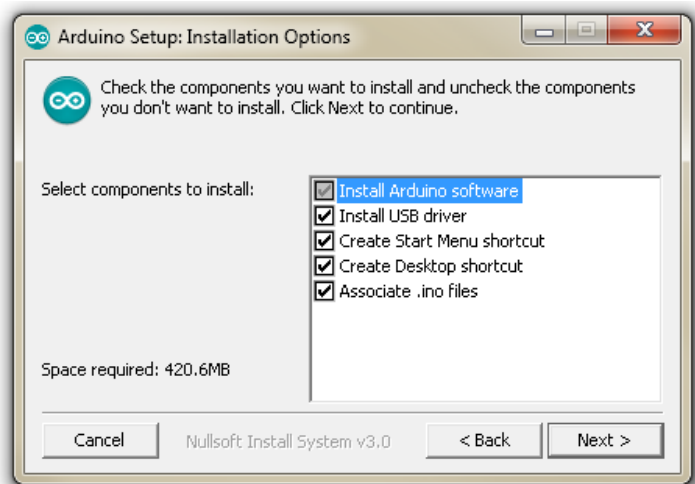


Gambar 2.2 File Arduino-1.8.5-windows.exe



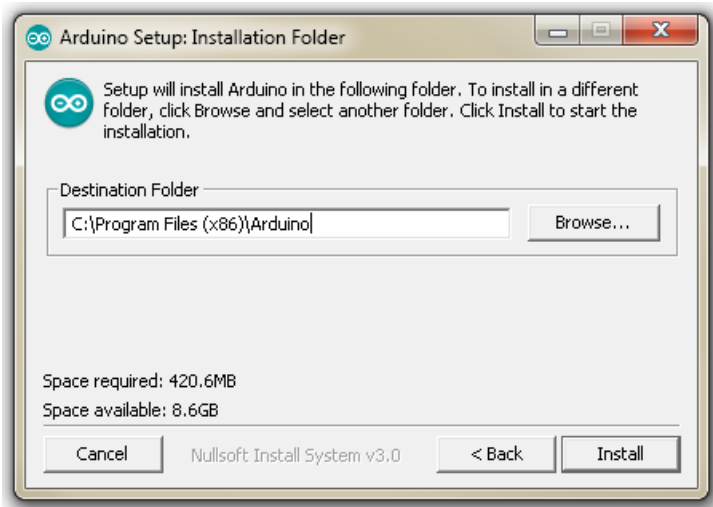
Gambar 2.3 Persetujuan instalasi aplikasi IDE Arduino

Dilanjutkan dengan menekan tombol **I Agree**, maka akan muncul jendela **Installation Option** seperti **Gambar 2.4**. Pastikan semua komponen terpilih/tercentang.



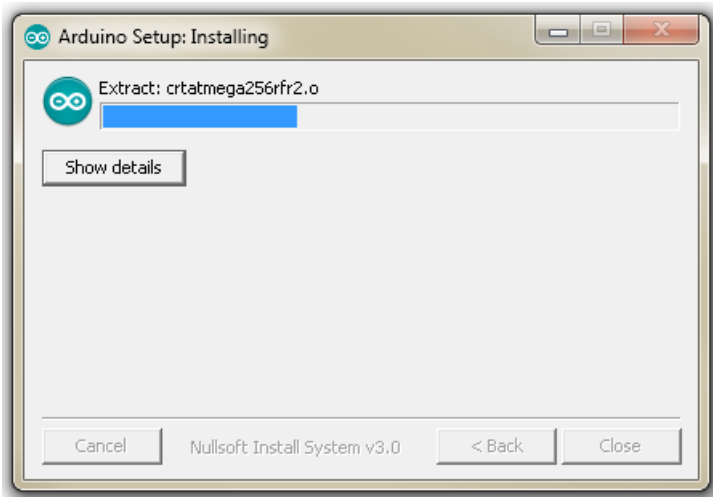
Gambar 2.4 Pilihan komponen instalasi

Tekan tombol Next, kemudian pilih Folder untuk menyimpan Aplikasi Arduino IDE.



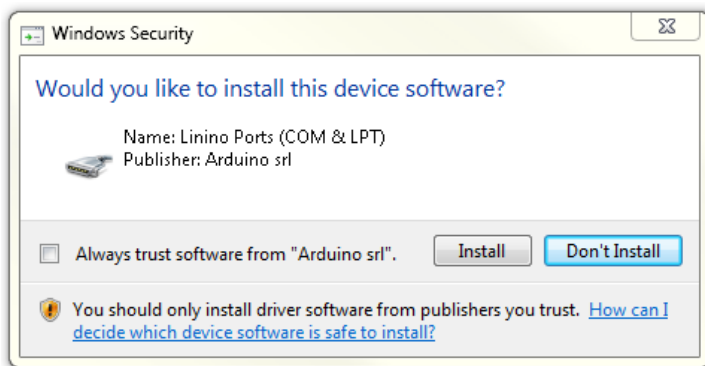
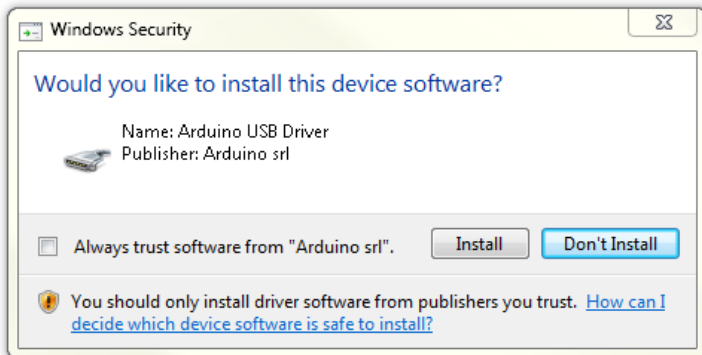
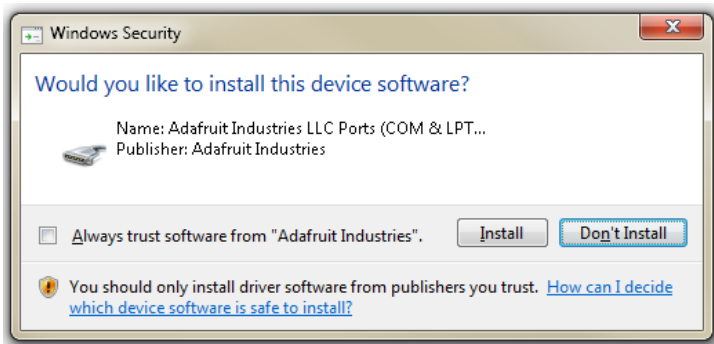
Gambar 2.5 Menentukan Folder instalasi

Tekan **Instal** untuk melanjutkan ke proses instalasi.



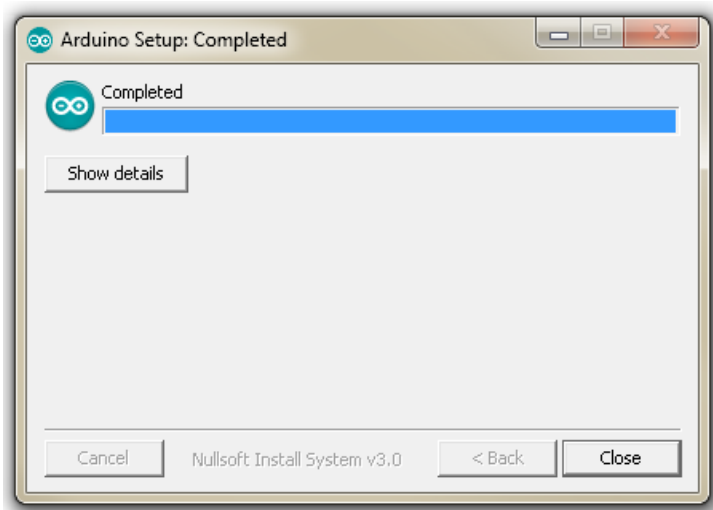
Gambar 2.6 Proses extract dan instalasi

Apabila muncul jendela seperti pada **Gambar 2.7**, tekan tombol **Instal** sampai instalasi selesai.



Gambar 2.7 Windows Security

Tekan **Close**, dan aplikasi Arduino IDE siap untuk digunakan.



Gambar 2.8 Instalasi selesai

Untuk membuka Arduino IDE, carilah file hasil instalasi **arduino.exe** kemudian double klik file tersebut.



Gambar 2.9 Aplikasi Arduino IDE

Aplikasi Arduino IDE Non-Installer

Untuk aplikasi Arduino IDE yang tidak memerlukan instalasi, dapat kita unduh juga di Website resmi Arduino yaitu <https://www.Arduino.cc/en/Main/Software> dan memilih **Windows ZIP file non admin install**. Ketika muncul pilihan **download and donate** dan **just download**, pilih just download maka secara otomatis file akan terunduh.



Gambar 2.10 File Arduino-1.8.5-windows.zip

Setelah file Arduino IDE terunduh, ekstrak file tersebut dengan cara klik kanan → Ekstrak disini/*Extract here*, maka hasilnya seperti pada **Gambar 2.11**.



Gambar 2.11 File Arduino.zip setelah diekstrak

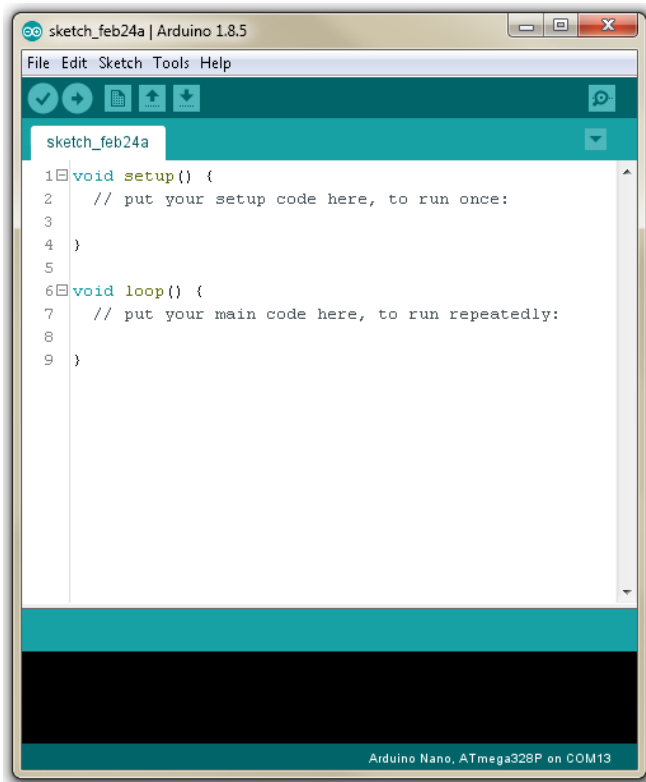
Untuk membuka aplikasi Arduino IDE, buka file **Arduino.exe** maka akan muncul seperti pada **Gambar 2.13**.



Gambar 2.12 Aplikasi Arduino IDE

2.2 Penggunaan Aplikasi Arduino IDE

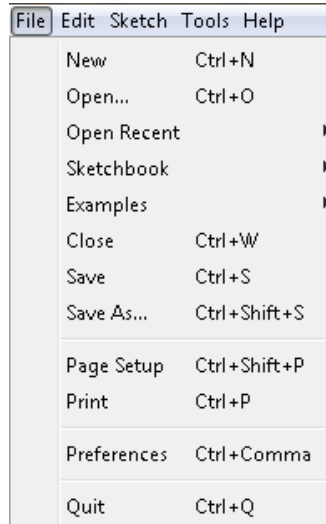
Setelah melakukan instalasi aplikasi Arduino IDE, selanjutnya akan dijelaskan tentang fungsi dari toolbar dan menu-menu yang ada di aplikasi Arduino IDE. **Gambar 2.13** menunjukkan tampilan utama dari aplikasi Arduino IDE.



Gambar 2.13 Tampilan utama aplikasi Arduino IDE v1.8.5

Berikut ini adalah penjelasan tentang menu-menu yang ada di Arduino IDE.

Menu File



Gambar 2.14 Menu File

- New, untuk membuat sketch yang baru;
- Open, untuk membuka file sketch yang tersimpan;
- Open Recent, untuk membuka file sketch yang baru dibuka;
- Sketchbook, untuk membuka file sketch yang pernah dibuat;
- Examples, untuk membuka contoh-contoh sketch yang disediakan oleh Arduino IDE;
- Close, untuk menutup sketch;
- Save, untuk menyimpan sketch;
- Save As, untuk menyimpan sketch dengan nama yang berbeda;
- Page Setup, untuk mengatur halaman ketika ingin mencetak sketch;
- Print, untuk mencetak sketch;
- Preferences, untuk membuka jendela preferences yang berisi pengaturan dari aplikasi Arduino IDE;
- Quit, untuk keluar dari Aplikasi Arduino IDE.

Menu Edit

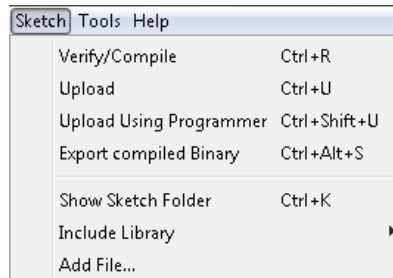


Gambar 2.15 Menu Edit

- Undo, untuk membatalkan beberapa perintah terakhir yang dilakukan;
- Redo, untuk mengembalikan suatu perintah yang telah dibatalkan menggunakan perintah Undo;
- Cut, untuk memindah text kode;
- Copy, untuk menggandakan/menyalin text kode;
- Copy for Forum, untuk menyalin text kode ke clipboard dalam bentuk yang sesuai dengan forum lengkap dengan pewarnaan sintaks;
- Copy as HTML, untuk menyalin teks kode ke clipboard sebagai HTML;

- Paste, untuk meletakkan text kode yang tersimpan di clipboard pada posisi kursor berada;
- Select All, untuk memilih semua text kode pada sketch;
- Go to line, untuk mengarahkan kursor ke baris sketch yang diinginkan;
- Comment/Uncomment, untuk memberikan atau menghilangkan komentar //di awal baris program yang ditentukan;
- Increase/Decrease Indent, untuk menambah/mengurangi indent atau tab pada awal baris yang dipilih;
- Increase/Decrease Font Size, untuk menaikkan/menurunkan ukuran huruf sketch;
- Find, Find Next, Find Previous, untuk membuka jendela Find dan Replace untuk mencari teks secara spesifik didalam suatu sketch.

Menu Sketch

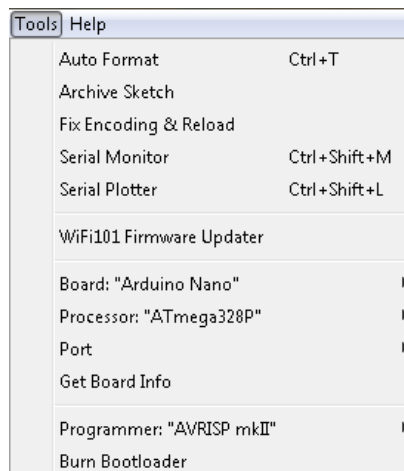


Gambar 2.16 Menu Sketch

- Verify/Compile, untuk mengecek kesalahan dari program yang dibuat;
- Upload, untuk mengecek dan memasukkan program ke IC Arduino;

- Upload Using Programmer, untuk menuliskan kembali bootloader pada board Arduino;
- Export compiled Binary, untuk menyimpan file .hex yang dapat disimpan sebagai arsip atau dikirim ke board dengan menggunakan alat lain;
- Show Sketch Folder, untuk menampilkan folder dari sketch yang sedang dibuka;
- Include Library, untuk menambahkan library program (Example pada menu file) baik secara online ataupun offline (.zip);
- Add File, untuk menambahkan sub sketch pada sketch utama.

Menu Tools

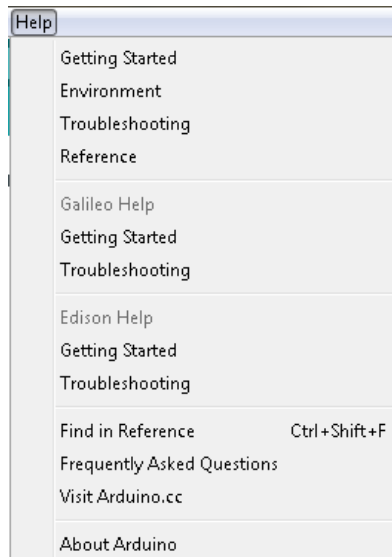


Gambar 2.17 Menu Tools

- Auto Format, untuk memperbaiki format sketch secara otomatis;
- Archive Sketch, untuk menyimpan sketch dalam bentuk file .zip;
- Fix Encoding & Reload, untuk membatalkan perubahan sketch dan mengambil ulang sketch sebelumnya sudah tersimpan;

- Serial Monitor, untuk menampilkan komunikasi serial antara Arduino dan komputer;
- Board, untuk memilih Board Arduino yang digunakan;
- Processor, untuk memilih jenis prosessor yang terpasang pada Arduino yang digunakan;
- Port, untuk memilih port serial yang digunakan dan tersambung pada Arduino;
- Programmer, untuk memilih jenis bootloader ketika memprogram sebuah board Arduino tidak menggunakan Serial Monitor;
- Burn Bootloader, untuk memasukkan bootloader pada mikrokontroler yang ada pada Arduino melalui pin ICSP.

Menu Help









Gambar 2.18 Menu Help

Menu ini digunakan untuk membantu pengguna menemukan akses mudah ke sejumlah dokumen Web yang disertakan pada Aplikasi

IDE Arduino. Pada menu ini memiliki akses ke Getting Started, Reference, Troubelshooting, panduan IDE dan dokumen lainnya tanpa koneksi internet.

Toolbar

	Verify untuk mengecek kesalahan dari program yang dibuat
	Upload untuk mengecek dan memasukkan program ke IC Arduino
	New untuk membuat sketch yang baru
	Open untuk membuka file sketch yang tersimpan
	Save untuk menyimpan sketch
	Serial Monitor untuk menampilkan komunikasi serial antara Arduino dan komputer

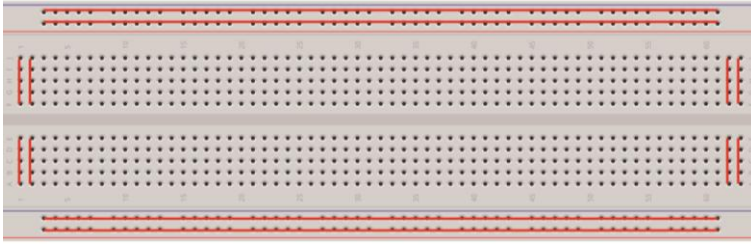
Project 1 - LED Berkedip

Pada bahasan ini kita akan mencoba langsung menggunakan Board Arduino untuk mengatur nyala dan matinya sebuah LED. Kita juga akan mempelajari dasar-dasar *input/output* dari sebuah Arduino. Arduino yang kita gunakan adalah Arduino Uno yang akan kita masukkan program melalui aplikasi Arduino IDE yang dapat kita *download* dengan mudah di internet.

Pada *project* ini akan mencoba untuk mengatur LED menjadi berkedip. Komponen yang harus dipersiapkan untuk memulai *project* ini adalah sebagai berikut.

Komponen	Jumlah
Papan Rangkaian	1
LED	1
Resistor 220 Ω	1
Arduino Uno	1
Kabel Penghubung	2

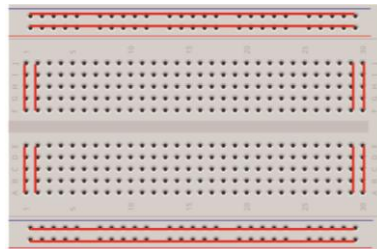
Dari komponen diatas, ada beberapa komponen yang akan sering kita gunakan untuk *project-project* selanjutnya, berikut ini penjelasan dari beberapa komponen diatas.



Gambar 3.1 Jalur Papan Rangkaian 830 titik

Pada papan rangkaian **Gambar 3.1** Jalur Papan Rangkaian 830 titik, terdapat jalur vertikal dan jalur horizontal. Untuk jalur vertikal terdapat 2 x 63 jalur dimana setiap jalur terdapat 5 titik yang saling terhubung (garis merah menunjukkan saling terhubung) yaitu jalur A1, B1, C1, D1, E1 dan jalur F1, G1, H1, I1, J1. Untuk jalur horizontal terdiri dari 4 jalur, dimana 2 jalur berada di atas dan 2 jalur berada di bawah. Pada setiap jalurnya terdapat 10 blok yang saling terhubung, yang mana setiap blok terdiri dari 5 titik.

Ada juga papan rangkaian yang ukurannya lebih kecil dari ukuran papan rangkaian di atas seperti ditunjukkan oleh **Gambar 3.2**.



Gambar 3.2 Jalur papan rangkaian – 400 titik

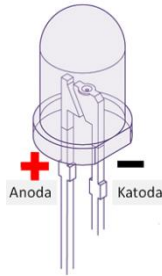
Pada papan rangkaian **Gambar 3.2**, terdapat jalur vertikal dan jalur horizontal. Untuk jalur vertikal terdapat 2 x 30 jalur dimana setiap jalurnya juga ada 5 titik yang saling terhubung yaitu jalur A1, B1, C1, D1, E1 dan jalur F1, G1, H1, I1, J1. Untuk jalur horizontal

terdiri dari 4 jalur, dimana 2 jalur berada diatas dan 2 jalur berada dibawah. Pada setiap jalurnya terdapat 5 blok yang saling terhubung, yang mana setiap blok terdiri dari 5 titik.

3.1 Light Emitting Diode (LED)

LED adalah salah satu komponen semikonduktor yang termasuk dalam jenis dioda. Sama-sama memiliki kutub positif dan kutub negatif, hanya saja LED memancarkan cahaya ketika diberikan tegangan dari anoda ke katoda. Hal yang perlu diperhatikan adalah cara mengetahui polaritas dari sebuah LED, berikut ini akan dijelaskan bagaimana mengetahui polaritas dari LED. Untuk mengatahuinya perhatikan kedua kaki LED, dimana kaki yang lebih panjang menunjukkan kutub positif (anoda) dan yang pendek adalah kutub negatif (katoda).

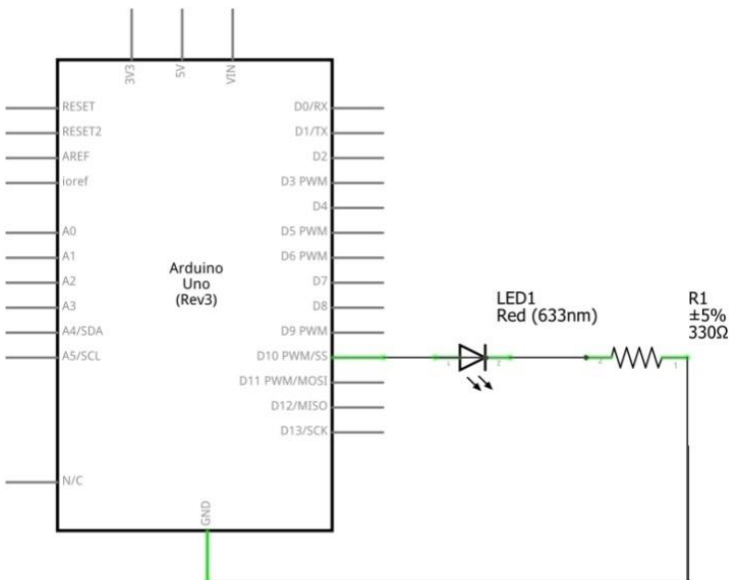
Selain itu, untuk melihat polaritasnya perhatikan lead frame kaki positif lebih kecil dibandingkan lead frame pada kaki negatif, dan ciri lainnya adalah kaki negatif terletak pada badan LED yang flat. Ada beberapa jenis warna LED, diantaranya adalah merah, kuning, hijau, biru, dan putih. Perbedaan warna tersebut terjadi karena perbedaan bahan semikonduktor yang digunakan. Saat ini teknologi LED memiliki banyak kelebihan seperti hemat listrik, tidak menimbulkan panas, lebih tahan lama, dan bentuknya yang kecil memudahkan dalam berbagai penggunaan.

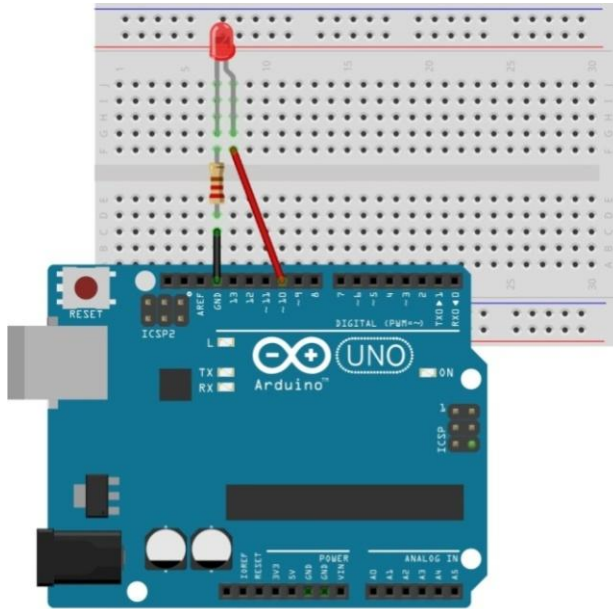


Gambar 3.3 *Light Emitting Diode (LED)*

3.2 Rangkaian Lampu Berkedip

Buatlah rangkaian seperti **Gambar 3.4**, untuk membuat rangkaian tersebut, hubungkan kaki negatif LED (kaki yang pendek/katoda) ke GND dengan disertai resistor 330 Ohm, sedangkan kaki LED positif (kaki yang panjang/anode) ke pin 10 Arduino. Untuk pemasangan kabel jumper, bisa dengan warna yang berbeda atau lubang yang berbeda, asalkan komponen dan kabel penghubung sesuai dengan gambar tersebut





Gambar 3.4 LED Berkedip

3.3 Program Lampu Berkedip

Berikut ini adalah sketch yang digunakan pada *project* ini sebagai awal dalam menggunakan Arduino.

```
int ledPin = 10;

void setup()
{
    //pin 10 sebagai OUTPUT
    pinMode(ledPin, OUTPUT);
}

void loop()
{
    digitalWrite(ledPin, HIGH);
    delay(1000);
    //LED hidup selama 1 detik
    digitalWrite(ledPin, LOW);
    delay(1000);
    //LED mati selama 1 detik
}
```

Baris yang dimulai dengan `//` atau `/*` dan diakhiri `*/` merupakan baris komentar dan tidak berpengaruh pada program yang dibuat. Kita dapat menuliskan apapun untuk menjelaskan sedikit code pada program kita atau memberikan informasi apapun yang membutuhkan penjelasan.

```
int ledPin = 10;
```

Kode pada baris tersebut digunakan untuk memberikan nama/inisialisasi pada pin output digital, yaitu pin 10 yang kita hubungkan ke kaki positif LED. Pin tersebut diberi nama `ledPin` dan merupakan variabel untuk menyimpan data. Jika kita ingin memberikan nama variabel untuk pertama kali, kita harus menjelaskan kepada compiler apa tipe variabel yang akan kita gunakan, seperti halnya pada kode diatas menggunakan tipe data integer.

```
void setup()
{
    //pin 10 sebagai OUTPUT
    pinMode(ledPin, OUTPUT);
}
```

Dalam program Arduino harus mempunyai fungsi `setup()` dan `loop()`, jika tidak maka tidak akan bekerja. Program Arduino diletakkan secara berurutan diantara 2 buah kurung kurawal `{}`. Tidak perlu khawatir dengan dua fungsi ini, karena Fungsi `setup()` dan `loop()` secara *default* akan muncul ketika aplikasi Arduino IDE dijalankan. Hal itu disebabkan karena ke dua fungsi tersebut sangatlah penting. Fungsi `setup()` berfungsi untuk inisialisasi mode pin sebagai *input* atau *output* dan inisialisasi serial.

```
void loop()
{
    digitalWrite(ledPin, HIGH);
    delay(1000);
    //LED hidup selama 1 detik
    digitalWrite(ledPin, LOW);
    delay(1000);
    //LED mati selama 1 detik
}
```

Fungsi `loop` akan terus bekerja secara terus menerus sampai Arduino dimatikan. Setiap pernyataan dalam fungsi `loop` (dalam kurung kurawal) dilakukan satu per satu, langkah demi langkah hingga bagian bawah fungsi tercapai, kemudian `loop` memulai lagi pada bagian atas fungsi, dan seterusnya sampai Arduino mati atau tekan tombol reset.

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

Menulis nilai `HIGH` pada `ledPin` (dalam hal ini pada digital pin 10), berarti mengatur digital pin ke `HIGH` dengan keluaran 5 V ke pin 10. Ketika mengatur digital pin ke `LOW` pin menjadi 0 V.

```
delay(1000);
```

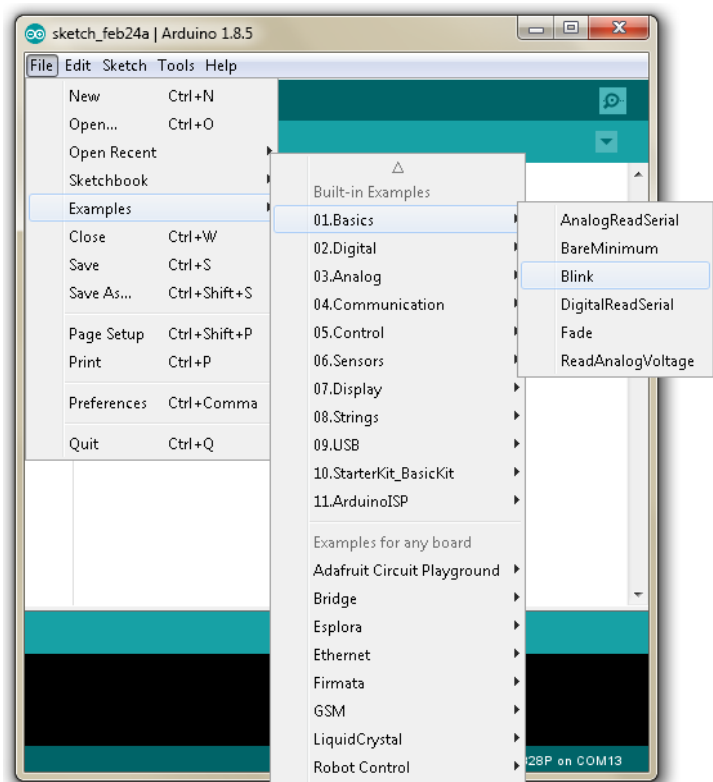
Pernyataan ini berfungsi untuk memberikan jeda selama 1000 milisekon untuk melaksanakan pernyataan berikutnya yaitu,

```
digitalWrite(ledPin, LOW);
```

Seperti yang telah disebutkan, ketika digital pin `LOW` atau 0, maka tegangan pada pin 10 yaitu 0 V yang artinya LED akan mati, diikuti dengan `delay(1000)` yang membuat LED menyala selama 1 detik. Ketika selesai, program itu akan diulang kembali dengan kondisi dimana `ledPin` `HIGH` atau LED hidup dalam waktu 1 detik, begitu seterusnya dan akan berhenti ketika Arduino dimatikan atau ditekan *reset*. Dengan mengikuti struktur program

langkah demi langkah lagi kita dapat melihat bahwa itu sangat sederhana.

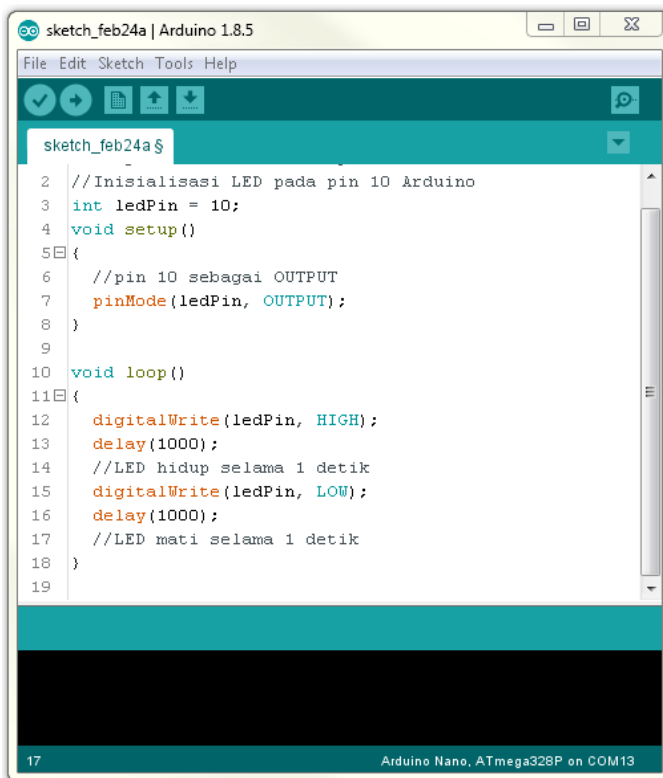
Program diatas dapat juga kita dapatkan pada library yang disediakan langsung di aplikasi IDE Arduino, caranya adalah buka **Menu File → 01.Basics → Blink (Gambar 3.5)** maka akan muncul sketch baru yang berisi program untuk membuat LED Berkedip/Blink.



Gambar 3.5 Membuka Contoh Sketch di Library Arduino IDE

3.4 Upload Program ke Board Arduino Uno

Selanjutnya akan dijelaskan tentang cara mengupload program ke board Arduino Uno. Pertama buka aplikasi Arduino IDE yang sudah kita unduh dan instal dengan cara klik double pada file **Arduino.exe**. Setelah terbuka, tuliskan program yang telah dijelaskan diatas pada aplikasi ArduinoIDE.



Gambar 3.6 Program LED Berkedip pada IDE

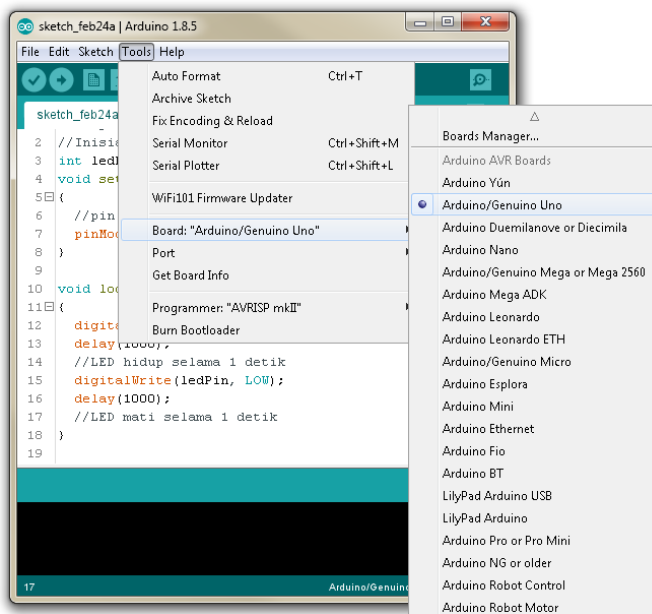
Setelah program ditulis, klik tombol **Compile** untuk mengecek apakah ada kesalahan atau tidak pada sketch. Apabila muncul notifikasi seperti **Gambar 3.7**, maka proses compiling berhasil dan tidak ada kesalahan pada sketch.


```
Done compiling.

Sketch uses 940 bytes (2%) of program storage space. Maximum is
Global variables use 9 bytes (0%) of dynamic memory, leaving 203
```

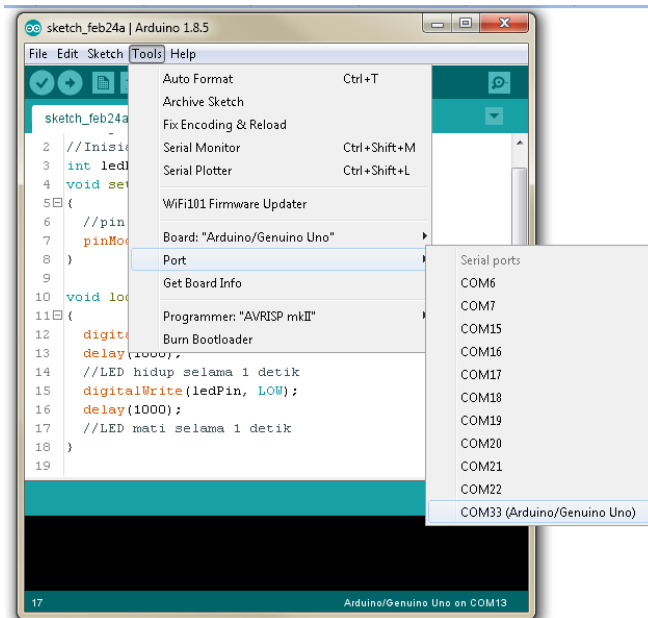
Gambar 3.7 Proses Compiling berhasil

Selanjutnya hubungkan Arduino Uno ke komputer/laptop. Atur jenis Board Arduino yang digunakan pada **Menu Tools → Board: “Jenis Arduino”** kemudian pilih **Arduino/Genuino Uno** (**Gambar 3.8**).



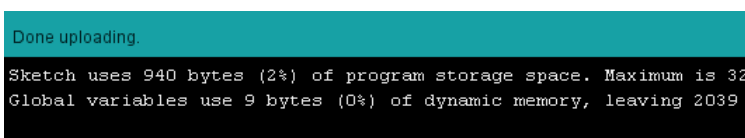
Gambar 3.8 Memilih Board Arduino di IDE

Selain itu pilih Port Arduino yang terhubung ke komputer/laptop (**Gambar 3.9**).



Gambar 3.9 Memilih Port Aduino di IDE

Sekarang tekan tombol **Upload** untuk memasukkan sketch ke board Arduino. Tunggu beberapa detik, jika kita perhatikan LED RX dan TX yang ada di board Arduino berkedip. Setelah beberapa detik akan muncul notifikasi “**Done uploading**” seperti pada **Gambar 3.10** yang menunjukkan proses Uploading berhasil.



Gambar 3.10 Proses Uploading berhasil

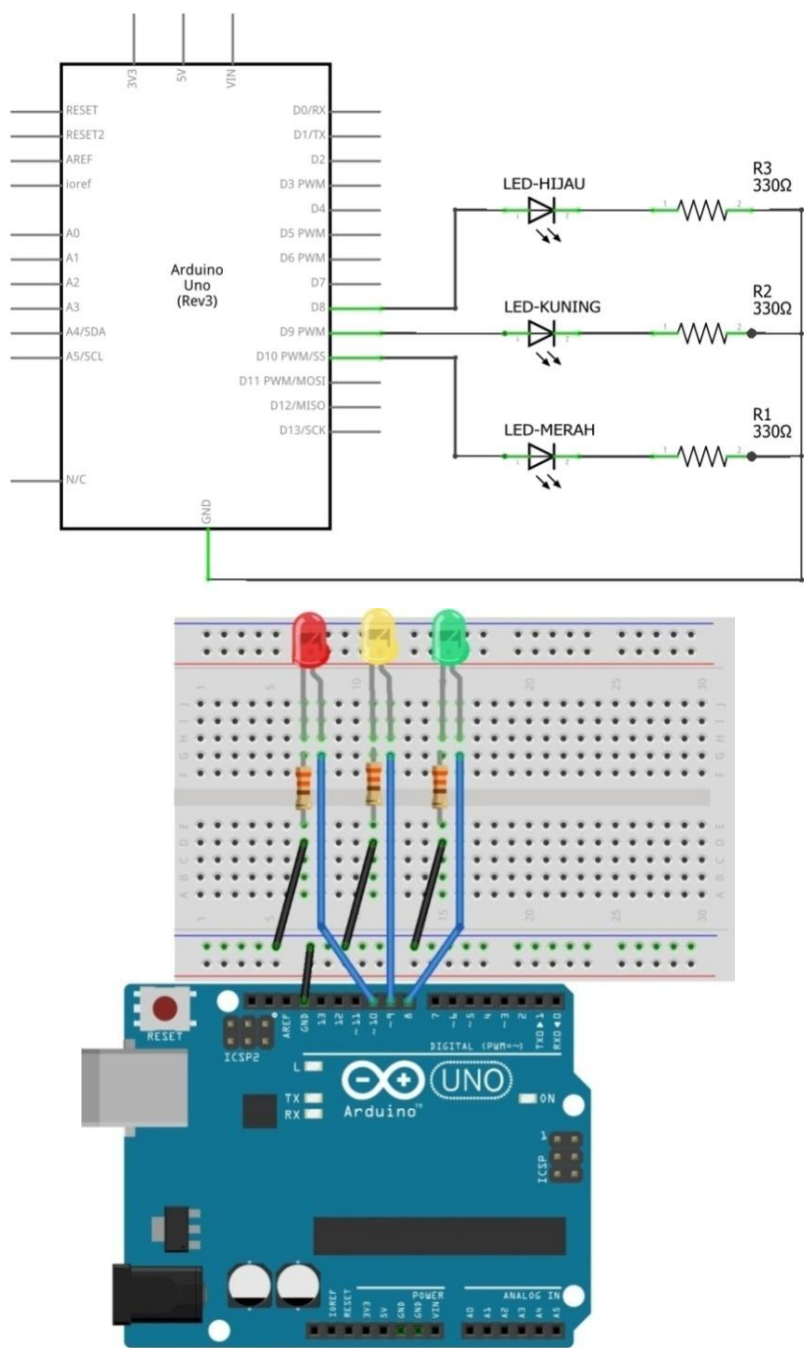
Demikian penjelasan tentang cara upload program ke board Arduino. Untuk *project* lainnya sama saja, hanya saja jika kita menggunakan board Arduino yang berbeda selain Arduino Uno, maka cukup ganti jenis Board yang digunakan.

Project 2 - LED Lampu Lalu Lintas

Pada pembahasan kali ini kita akan membahas salah satu aplikasi sederhana pada Arduino dan LED untuk digunakan sebagai lampu lalu lintas atau kita sering menyebutnya lampu merah. Berbeda dengan *project* sebelumnya, *project* ini menggunakan 3 buah LED berwarna merah, kuning dan hijau sebagai indikator, dimana merah berarti berhenti, kuning berarti bersiap-siap, dan hijau berarti jalan. Prinsip kerja lampu lalu lintas sendiri sebenarnya hanya seperti LED berjalan, namun untuk lampu lalu lintas sedikit lebih kompleks dan perlu pengaturan waktu. Komponen yang harus disiapkan adalah sebagai berikut.

Komponen	Jumlah
Papan Rangkaian	1
LED (Merah, Kuning, Hijau)	3
Resistor 220 Ω	3
Arduino Uno	1
Kabel Penghubung	7

4.1 Rangkaian Simulasi Lampu Lalu Lintas



Gambar 4.1 Rangkaian Traffic Light

Buatlah rangkaian seperti pada **Gambar 4.1**.

- Pasangkan kabel jumper dari pin GND Arduino ke bagian bawah papan rangkaian
- Hubungkan ketiga kaki negatif LED ke GND dengan disertai resistor 330 Ohm.
- Hubungkan kaki positif LED Merah ke pin 10 Arduino
- Hubungkan kaki positif LED Kuning ke pin 9 Arduino
- Hubungkan kaki positif LED Hijau ke pin 8 Arduino
- Untuk pemasangan kabel jumper, bisa dengan warna yang berbeda atau lubang yang berbeda, asalkan komponen dan kabel penghubung sesuai dengan gambar diatas.

4.2 Program Simulasi Lampu Lalu Lintas

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
//inisialisasi pin
int ledDelay = 10000;
int redPin = 10;
int yellowPin = 9;
int greenPin = 8;

void setup() {
    pinMode(redPin, OUTPUT);
    pinMode(yellowPin, OUTPUT);
    pinMode(greenPin, OUTPUT);
}

void loop() {
    //lampu merah hidup selama 10 detik
    digitalWrite(redPin, HIGH);
    delay(ledDelay);
```

```

//lampu kuning hidup selama 2 detik
digitalWrite(yellowPin, HIGH);
delay(2000);

//lampu hijau hidup, lampu merah dan kuning mati
//berlangsung selama 10 detik
digitalWrite(greenPin, HIGH);
digitalWrite(redPin, LOW);
digitalWrite(yellowPin, LOW);
delay(ledDelay);

//lampu kuning hidup dan lampu hijau mati
//berlangsung selama 2 detik
digitalWrite(yellowPin, HIGH);
digitalWrite(greenPin, LOW);
delay(2000);
//lampu kuning mati
digitalWrite(yellowPin, LOW);
//kembali ke program awal
}

```

Program tersebut diawali dengan lampu merah hidup selama 10 detik yang artinya pengendara dilarang melintas, kemudian diikuti dengan lampu kuning menyala selama 2 detik yang menandakan pengendara bersiap-siap untuk berjalan.

```

digitalWrite(redPin, HIGH);
delay(ledDelay);
digitalWrite(yellowPin, HIGH);
delay(2000);

```

Setelah itu lampu hijau hidup dengan lampu merah dan kuning mati selama 10 detik yang artinya pengendara boleh melintas.

```

digitalWrite(greenPin, HIGH);
digitalWrite(redPin, LOW);
digitalWrite(yellowPin, LOW);
delay(ledDelay);

```

Dilanjutkan dengan lampu kuning hidup dan lampu hijau mati berlangsung selama 2 detik.

```
digitalWrite(yellowPin, HIGH);  
digitalWrite(greenPin, LOW);  
delay(2000);
```

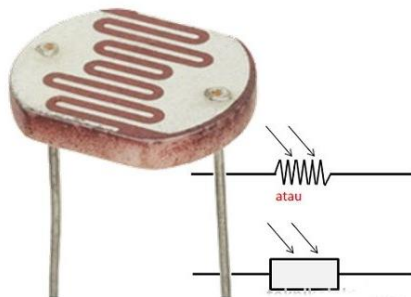
Kemudian lampu kuning mati dan terakhir program di ulang kembali ketika lampu merah hidup.

```
digitalWrite(yellowPin, LOW);
```

Dengan mengikuti struktur program langkah demi langkah lagi kita dapat memahami cara kerja dari lampu merah yang selama ini kita lihat di persimpangan jalan. Untuk mendapatkan hasil yang maksimal silahkan uji coba program diatas dan gunakan waktu yang sesuai dengan lampu merah sesungguhnya.

Project 3 - Sensor Cahaya (LDR)

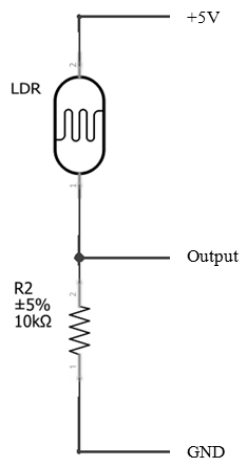
Light Dependent Resistor atau LDR atau disebut juga Photoresistor merupakan salah satu jenis reistor dimana besarnya hambatan/resistansinya berdasarkan intensitas cahaya yang diterima. Nilai hambatan dari LDR akan besar apabila intensitas cahaya yang diterimanya rendah. Hambatan dari LDR akan kecil apabila instensitas cahaya yang diterima tinggi. Artinya nilai hambatan dari LDR berbanding terbalik dengan intensitas cahaya yang diterimanya. Dalam kondisi terang (instensitas cahaya tinggi), nilai hambatan dari LDR dapat mencapai 200 Kilo Ohm, sedangkan pada kondisi gelap (intensitas cahaya rendah) nilai hambatannya menurun menjadi 500 Ohm.



Gambar 5.1 Bentuk dan simbol LDR

LDR dapat diaplikasikan pada beberapa rangkaian elektronik, LDR banyak digunakan pada lampu penerangan jalan sebagai pendeteksi/sensor cahaya. Pada kondisi terang, lampu akan mati dan akan hidup kembali ketika kondisi malam hari atau kondisi gelap. Tidak hanya itu, LDR juga biasa digunakan pada shutter kamera, alarm, lampu kamar tidur dan lain sebagainya.

Dalam menggunakan LDR dalam sebuah rangkaian menggunakan prinsip pembagi tegangan seperti pada **Gambar 5.2**.



Gambar 5.2 Rangkaian pembagi tegangan LDR

Ketika kondisi gelap, tegangan yang melewati LDR akan menjadi lebih rendah disebabkan karena resistansi LDR besar, dan tegangan akan lebih tinggi ketika kondisi terang karena resistansi LDR kecil. Perbedaan tegangan tersebut akan digunakan Arduino untuk memberikan respon pada rangkaian lainnya, dalam hal ini akan digunakan untuk menghidupkan LED.

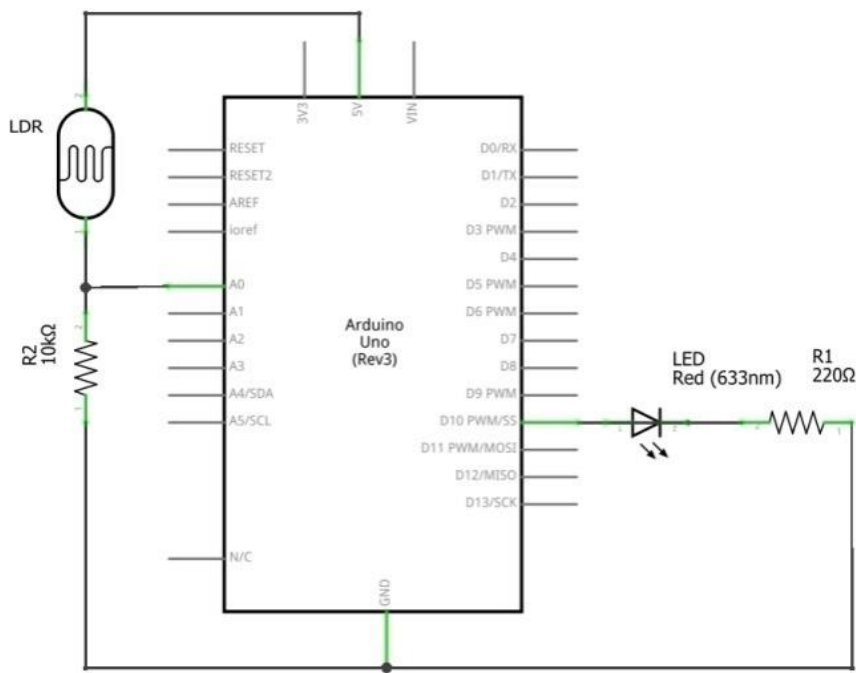
Pada *project* ini, kita akan memanfaatkan fungsi dari LDR itu sendiri untuk dijadikan sebagai saklar LED otomatis. Ketika

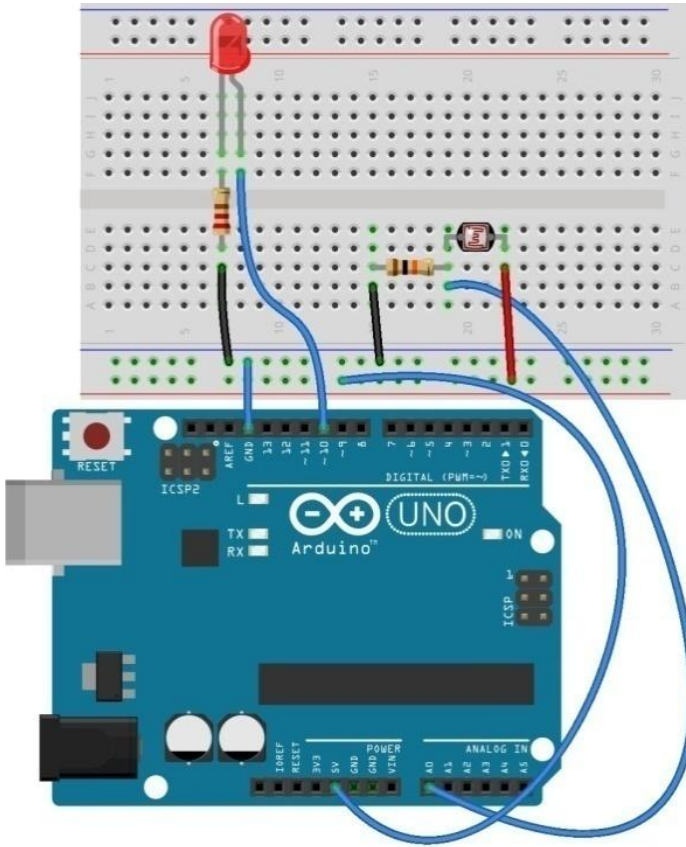
kondisi gelap/intensitas cahaya rendah, LED akan hidup. Ketika kondisi terang/intensitas cahaya tinggi, LED akan mati. Untuk membuat *project* tersebut, berikut ini adalah komponen yang dibutuhkan.

Komponen	Jumlah
LED	1
Resistor 220 Ω	1
Resistor 10 k Ω	1
Sensor LDR	1
Arduino Uno	1
Papan Rangkaian	1
Kabel Penghubung	7

5.1 Rangkaian Sensor Cahaya

Buatlah rangkaian seperti **Gambar 5.3** berikut ini.





Gambar 5.3 Rangkaian sensor cahaya

5.2 Program Sensor Cahya

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
//Inisialisasi pin
int ledPin = 13;
int ldrPin = A0;

void setup() {
Serial.begin(9600);
pinMode(ledPin, OUTPUT);
```

```
pinMode(ldrPin, INPUT);
}

void loop() {
    //membaca nilai input analog
    int ldrStatus = analogRead(ldrPin);
    //pengujian kondisi
    if (ldrStatus <= 200)
    {
digitalWrite(ledPin, HIGH);
Serial.print("Gelap, LED hidup : ");
Serial.println(ldrStatus);
    }
    else
    {
digitalWrite(ledPin, LOW);
Serial.print("Terang, LED mati : ");
Serial.println(ldrStatus);
    }
}
```

Seperti *project* sebelumnya, program ini diawali dengan inisialisai variabel yang yaitu ledPin pada pin *output* 13 dan ldrPin pada pin analog A0.

```
void setup() {
Serial.begin(9600);
pinMode(ledPin, OUTPUT);
pinMode(ldrPin, INPUT);
}
```

Pada void setup merupakan bagian untuk inisialisasi mode pin sebagai *input* dan *output*. Serial.begin berfungsi untuk membuka port data serial untuk komunikasi serial baik mengirim atau menerima data dari serial. Nilai 9600 merupakan baud rate yang digunakan untuk komunikasi serial.

```
int ldrStatus = analogRead(ldrPin);
```

Kode tersebut digunakan untuk membaca *input* analog dengan resolusi 10 bit, dalam hal ini diinisialisasikan sebagai `ldrStatus`.

```
if (ldrStatus <= 200)
{
digitalWrite(ledPin, HIGH);
Serial.print("Gelap, LED hidup : ");
Serial.println(ldrStatus);
}
```

Dilanjutkan dengan pengujian kondisi, jika *input* analog bernilai kurang dari sama dengan 200 maka LED akan hidup (kondisi gelap). Apabila kita membuka jendela serial monitor akan muncul Gelap, LED hidup : **“nilai analog”**.

```
else
{
digitalWrite(ledPin, LOW);
Serial.print("Terang, LED mati : ");
Serial.println(ldrStatus);
}
```

Jika *input* analog lebih besar dari 200 maka LED akan mati (kondisi terang), dan pada jendela serial monitor akan muncul Terang, LED mati : **“nilai analog”**.

Fungsi yang ada didalam loop akan terus bekerja secara terus menerus sampai Arduino dimatikan. Setiap pernyataan dalam fungsi loop (dalam kurung kurawal) dilakukan satu per satu, langkah demi langkah hingga bagian bawah fungsi tercapai, kemudian loop memulai lagi pada bagian atas fungsi, dan seterusnya sampai Aduino mati atau tekan tombol reset.

Project 4 - Sensor Gerak (PIR)

Passive Infra-Red atau PIR merupakan sensor yang dapat mendeteksi pergerakan dengan sistem deteksi pancaran sinar infrared. Prinsip kerja sensor PIR didasarkan pada ada tidaknya perubahan suhu sekarang dan sebelumnya. Pancaran dari sinar infra-red akan mengenai sensor Pyroelektrik dan akan dihasilkan arus listrik akibat sinar infra-red memiliki energi kalor. Arus tersebut akan menghasilkan tegangan dan akan masuk ke IC komparator, kemudian tegangan tersebut akan dibandingkan dengan tegangan referensi/ V_{ref} dan akan menghasilkan output sinyal sebesar 1 bit. Sehingga output dari sensor PIR ini berupa nilai *High* atau *Low*.

Sensor PIR dapat mendeteksi pancaran Infra-Red dengan panjang gelombang berkisar 8-14 μm , sedangkan manusia memiliki suhu badan yang dapat menghasilkan pancaran Infra-Red dengan panjang gelombang berkisar 9-10 μm , untuk itu pancaran tersebut digunakan sensor PIR untuk mendeteksi pergerakan manusia. Contoh penggunaan sensor ini adalah pintu Mall yang dapat membuka secara otomatis.



Gambar 6.1 Sensor PIR

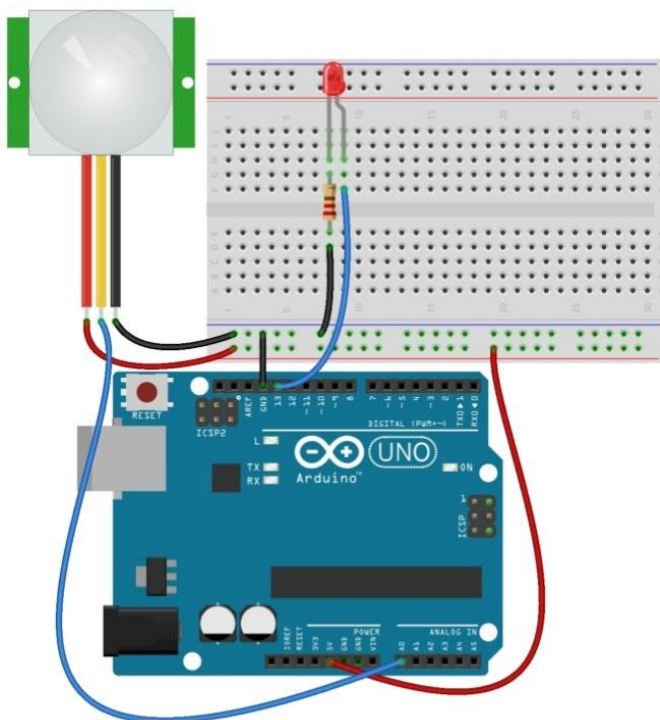
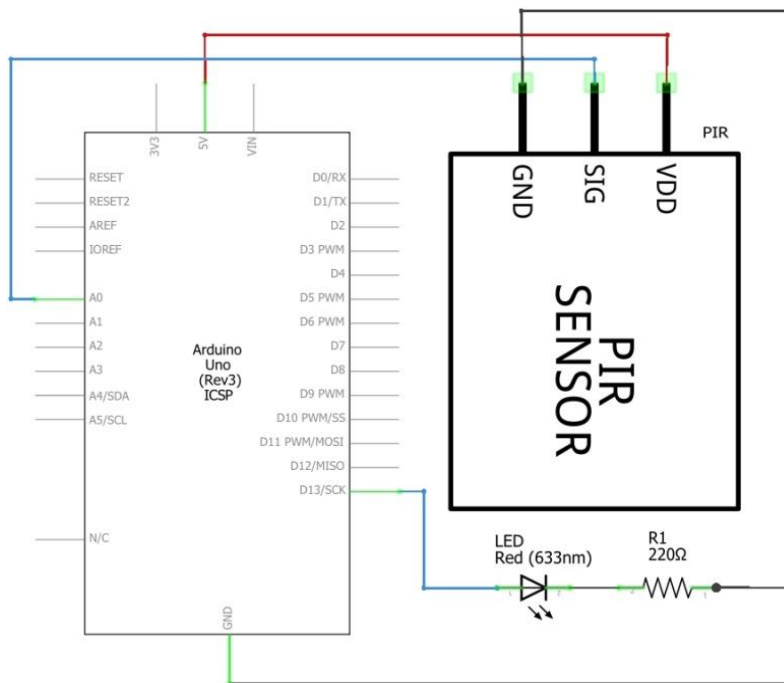
Sensor PIR bekerja pada tegangan 5V dan memiliki 3 pin yaitu GND, Output, dan VCC. Output tersebut akan dimasukkan ke pin input analog yang ada di board Arduino.

6.1 Rangkaian Sensor Gerak

Pada *project* kali ini akan dibuat sebuah alat untuk mendeteksi pergerakan dengan menggunakan sensor PIR dan outputnya berupa LED sebagai beringatan apabila ada pergerakan. Berikut ini adalah komponen yang dibutuhkan dalam membuat *project* ini.

Komponen	Jumlah
Papan Rangkaian	1
Arduino Uno	1
LED Merah	1
Resistor 220 Ω	1
Kabel Penghubung	7

Buatlah rangkaian seperti **Gambar 6.2**.



Gambar 6.2 Rangkaian Sensor Gerak

6.2 Program Sensor Gerak

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
//Inisialisasi pin
int ledPin = 13;
int inputPin = A0;
int pirState = LOW;
int val = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  val = digitalRead(inputPin);

  if (val == HIGH)
  {
    digitalWrite(ledPin, HIGH);
    if (pirState == HIGH) {
      Serial.println("Gerakan Terdeteksi!");
      pirState = LOW;
    }
  }

  else
  {
    digitalWrite(ledPin, LOW);
    if (pirState == LOW){
      Serial.println("Tidak Ada Gerakan!");
      pirState = HIGH;
    }
  }
}
```

Program tersebut dimulai dengan inisialisasi variabel yang digunakan.

```
int ledPin = 13;
int inputPin = A0;
int pirState = LOW;
int val = 0;
```

`ledPin` digunakan sebagai pin *output* ke LED pada pin 13. `inputPin` adalah inisial pin *input* dari sensor PIR. `pirState=LOW` merupakan asumsi yang kita berikan saat tidak ada pergerakan.

```
void loop()
{
    val = digitalRead(inputPin);

    if (val == HIGH)
    {
        digitalWrite(ledPin, HIGH);
        if (pirState == HIGH) {
            Serial.println("Gerakan Terdeteksi!");
            pirState = LOW;
        }
    }
}
```

Arduino membaca `inputPin` dalam bentuk HIGH/LOW yang diinisialkan sebagai `val`. Ketika `val` bernilai HIGH atau terdeteksi adanya pergerakan, maka LED akan hidup. Tidak hanya itu, ketika `pirState` yang kita atur default LOW berubah menjadi HIGH maka akan muncul peringatan “Gerakan Terdeteksi!”. Peringatan ini muncul hanya sekali saja, dan akan berubah ketika terjadi perubahan pada `pirState`.

```
else
{
    digitalWrite(ledPin, LOW);
    if (pirState == LOW){
```

```
    Serial.println("Tidak Ada Gerakan!");  
    pirState = HIGH;  
  }  
}
```

Ketika `val` bernilai `LOW`, maka LED akan mati dan `pirState` masih dalam keadaan default sehingga akan muncul notifikasi “Tidak Ada Gerakan”. Notifikasi tersebut akan berubah ketika terjadi perubahan pada `pirState`.

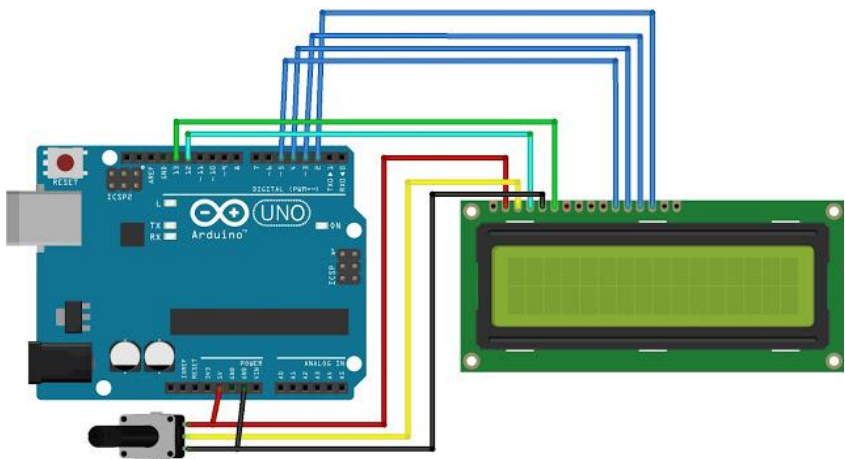
Project 5 - Module LCD 16x2 dengan I2C Konverter

7.1 Modul LCD 16x2 dengan I2C Konverter

Pada bahasan sebelumnya hasil pengukuran ditampilkan di Serial Monitor IDE Arduino, maka pada bahasan ini akan ditampilkan menggunakan LCD 16x2 yang sudah dilengkapi dengan modul I2C Konverter. Modul ini berguna untuk memudahkan kita dalam menggunakan LCD dimana pin yang dipakai hanyalah 4 saja.

I2C adalah modul LCD yang dikendalikan secara serial sinkron dengan protokol I2C/IIC (*Inter Integrated Circuit*) atau TWI (*Two Wire Interface*). Normalnya, modul LCD dikendalikan secara parallel baik untuk jalur data maupun kontrolnya. Namun, jalur parallel akan memakan banyak pin di sisi kontroler. Modul LCD 16x2 mempunyai 16 pin interface yaitu VSS VDD V0 RS R/W E D0 D1 D2 D3 D4 D5 D6 D7 BLA BLK dimana 6 atau 7 pin adalah pin digital. Berikut adalah rangkaian Modul LCD 16x2 tanpa modul I2C. Untuk sebuah controller yang banyak mengendalikan banyak I/O, menggunakan jalur parallel adalah solusi yang kurang tepat. Sebagai contoh, sebuah Arduino UNO

memiliki pin digital sebanyak 13 buah. Jika digunakan separuhnya untuk mengendalikan LCD berarti hanya ada alternatif pin sekitar 6 atau 7 lagi untuk mengendalikan perangkat lainnya. Untuk itu perlu digunakan jalur serial (I2C) menggunakan modul I2C konverter.



Gambar 7.1 Rangkaian Display LCD 16x2

Arduino sendiri sudah mendukung protokol I2C/IIC. Di papan Arduino UNO, Port I2C terletak pada pin A4 untuk jalur SDA (Serial Data) dan pin A5 untuk SCL (Serial Clock), dan tidak lupa menghubungkan GND dan VCC.



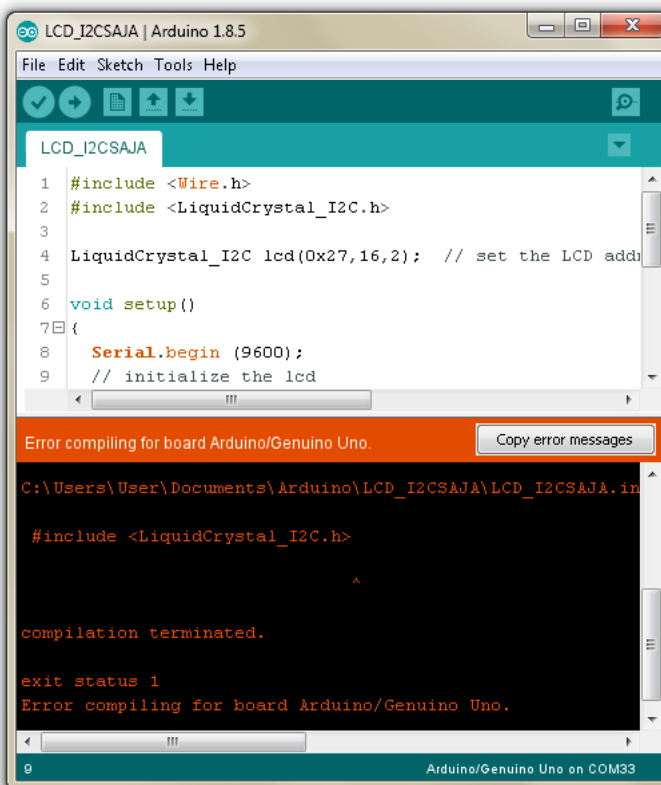
Gambar 7.2 Modul LCD 16x2 dengan I2C Konverter

Modul I2C konverter menggunakan chip IC PCF8574 produk dari NXP sebagai kontroler. IC yang digunakan ini adalah 8 bit I/O expander untuk I2C bus yang berkerja seperti shift register. I2C

LCD mempunyai 16 pin Output yang bisa dihubungkan dengan pin LCD 16x2 secara langsung (disolder permanen) dan memiliki 4 pin input (VCC, GND, SDA, SCL). Pada *project* kali ini kita menggunakan modul I2C Konverter dengan tipe IC PCF8574T dengan alamat LCD Address 16 kata dan 2 baris yaitu 0x27.

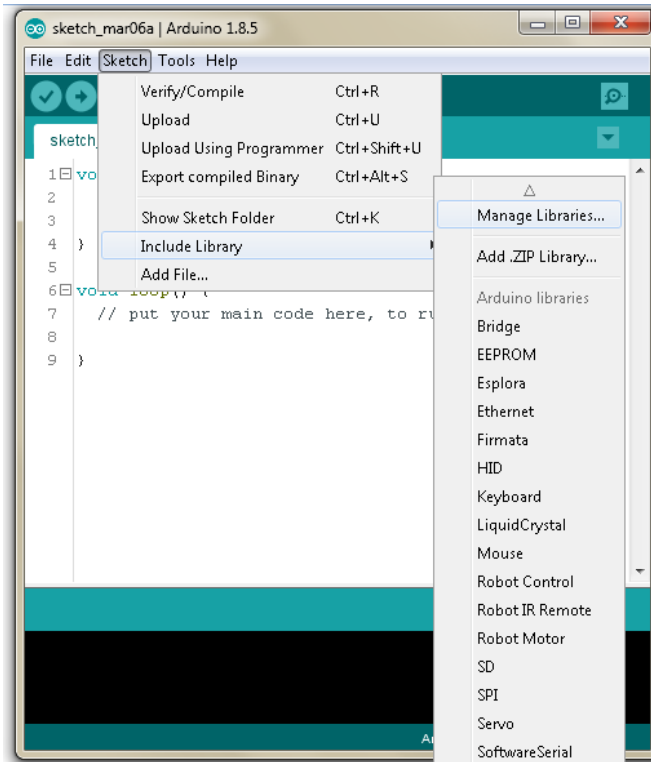
7.2 Menambahkan Library IDE Arduino

Penggunaan module Arduino pada sebuah project, contohnya modul LCD 16x2 dengan I2C Konverter, ada kalanya aplikasi IDE Arduino akan mendeteksi kesalahan pada program setelah kita lakukan *Compile*, seperti terlihat pada **Gambar 7.3**.

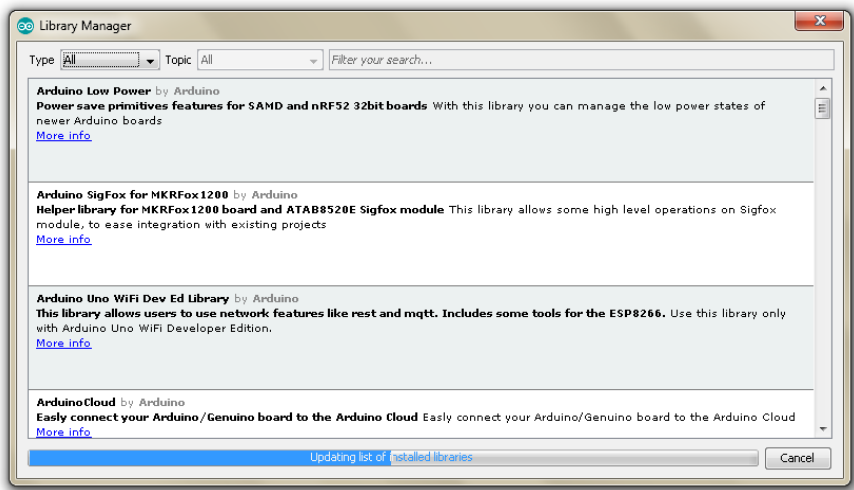


Gambar 7.3 Library tidak ditemukan pada IDE Arduino

Pada **Gambar 7.3** tersebut terlihat *Library* dari `LiquidCrystal_I2C.h` belum ada di IDE Arduino, sehingga muncul error saat di *Compile*. Untuk itu kita perlu menambahkan *Library* untuk modul yang kita gunakan baik secara online maupun offline pada IDE Arduino. Masuk pada Menu **Sketch** → **Include Library**, maka akan muncul pilihan yaitu **Manage Libraries** (menambahkan *Library* secara online) dan **Add .ZIP Library** (menambahkan *Library* secara offline). Kita akan mencoba menambahkan *Library* secara online terlebih dahulu, maka akan muncul jendela seperti pada **Gambar 7.5**.

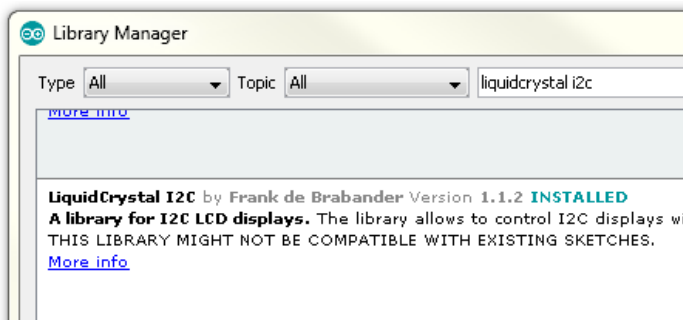


Gambar 7.4 Proses menambahkan Library IDE Arduino



Gambar 7.5 Jendela *Library Manager*

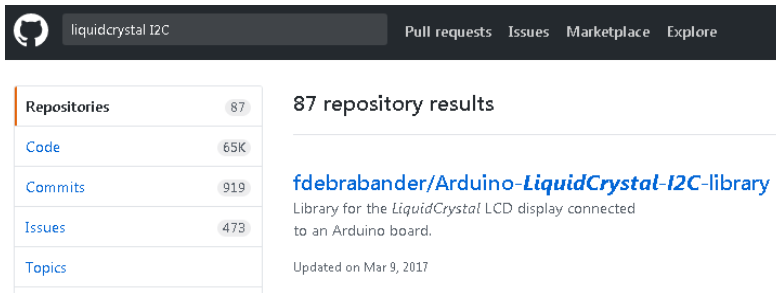
Pastikan komputer/laptop yang digunakan sudah terhubung dengan internet, maka secara otomatis aplikasi akan melakukan *Updating* terlebih dahulu. Kemudian tulis pada kolom pencarian untuk mencari *library* yang kita butuhkan, dalam hal ini kita akan menambahkan *Library LiquidCrystal I2C*.



Gambar 7.6 Proses pencarian *Library LiquidCrystal I2C*

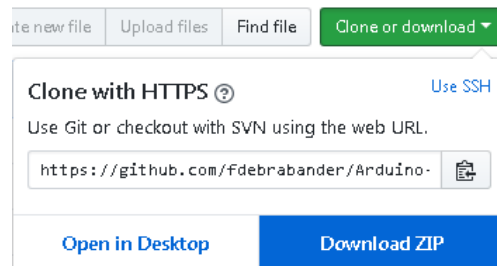
Setelah melakukan pencarian, pilih *LiquidCrystal I2C* by *Frank de Brabander*, tentukan *Version* kemudian tekan *Instal*, maka *Library* tersebut akan terdownload secara otomatis.

Untuk menambahkan *Library* secara offline, terlebih dahulu kita sudah memiliki *file Library .ZIP*, apabila belum memilikinya dapat kita download di website yang direkomendasikan oleh IDE seperti <https://github.com/>. Cukup tuliskan *Library* yang kita butuhkan pada kolom pencarian, seperti ***LiquidCrystal I2C***, maka akan muncul beberapa pilihan *Library* seperti pada **Gambar 7.7**.



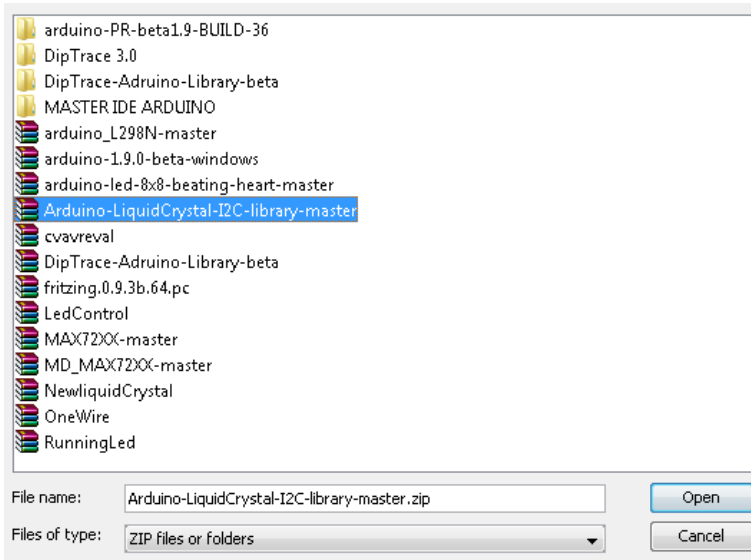
Gambar 7.7 Proses pencarian *Library* di website github.com

Pilih ***fdebrabander/Arduino-LiquidCrystal-I2C-library*** maka akan diarahkan pada halaman berikutnya.



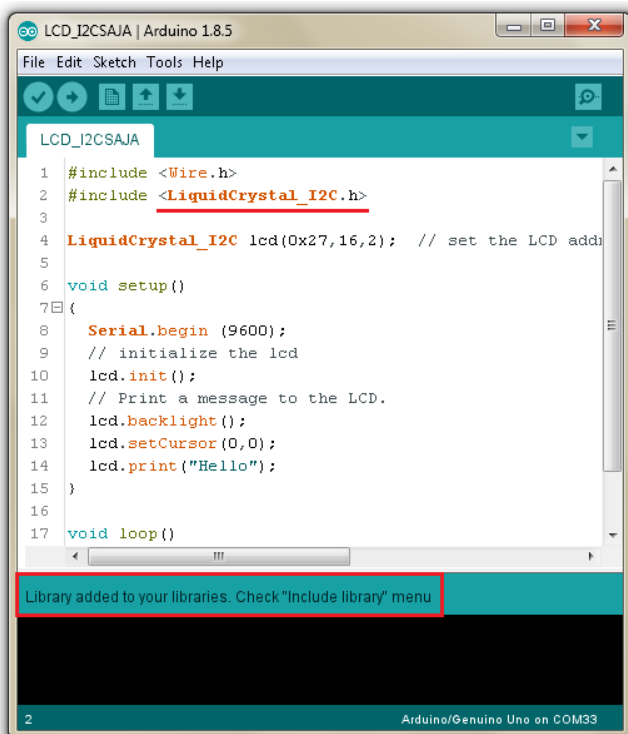
Gambar 7.8 Proses download *Library* di website github.com

Pilih pada ***Clone or download*** dan pilih ***Download ZIP***, maka secara otomatis browser akan mendownloadnya secara otomatis. Setelah proses download selesai, masuk pada Menu ***Sketch*** → ***Include Library*** dan pilih ***Add .ZIP Library***. Kemudian cari file ***.ZIP*** yang sudah didownload.



Gambar 7.9 Proses menambahkan file .ZIP

Pilih *File Library* kemudian tekan ***Open***.



Gambar 7.10 Proses penambahan *Library* berhasil

Gambar 7.10 menunjukkan penambahan *Library* secara offline berhasil dan kode sketch inisialisasi yang awalnya berwarna hitam berubah menjadi warna merah yang menunjukkan *Library LiquidCrystal I2C* terdeteksi.

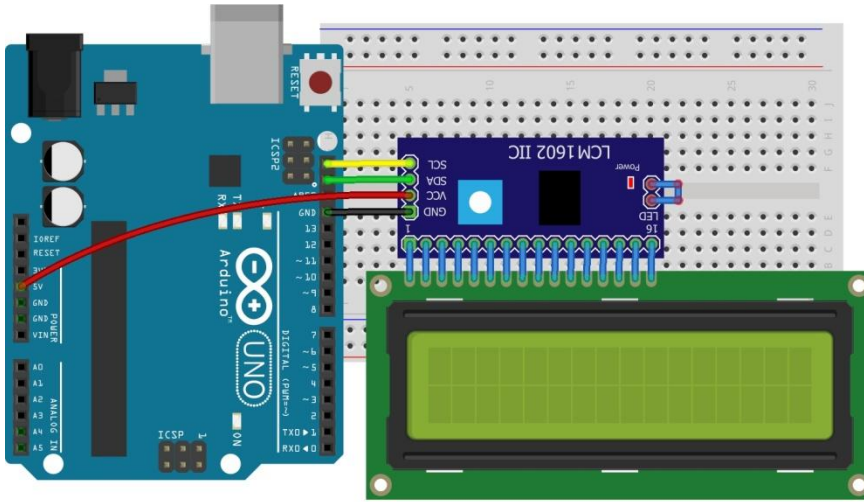
Untuk penambahan *library* modul lainnya, cukup lakukan hal yang sama dengan mengganti keyword pencarian sesuai dengan modul Arduino yang digunakan baik secara online atau offline. Pilihlah *library* yang sesuai dengan nama, jenis, dan tipe IC yang digunakan modul tersebut.

7.3 Rangkaian LCD 16x2 dengan I2C Konverter

Berikut ini adalah komponen yang kita butuhkan untuk memulai project ini.

Komponen	Jumlah
Arduino Uno	1
LCD 16x2 dengan modul I2C Konverter	1
Kabel Jumper	4

Buatlah rangkaian seperti pada **Gambar 7.11** berikut ini



Gambar 7.11 Rangkaian Modul LCD 16x2 dengan I2C Konverter

Dimana pin SCL dan SDA Arduino terhubung dengan pin SCL dan SDA modul I2C. Untuk GND dan VCC modul I2C terhubung ke VCC 5V dan GND Arduino.

7.4 Program

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);

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

void loop()
{

```

```

    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print("Hello");
    lcd.setCursor(0,1);
    lcd.print("LCD I2C TEST");
}

```

Penggunaan modul I2C konverter ini dibantu dengan adanya *Library* `Wire.h` yang sudah tersedia di IDE Arduino. *Library* ini berfungsi untuk mengkonversi jalur paralel LCD menjadi jalur serial I2C. Selain itu juga kita dapat menggunakan *Library* dari `LiquidCrystal_I2C.h` agar lebih memudahkan kita dalam membuat sketch, sehingga kita cukup mendefinisikan sebagai berikut.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

```

Kedua library tersebut wajib ada ketika kita ingin menggunakan LCD 16x2 dengan modul I2C Konverter. Perhatikan kode berikut ini.

```

LiquidCrystal_I2C lcd(0x27,16,2);

```

Kode tersebut adalah alamat untuk modul I2C yang kita gunakan dengan tipe IC PCF8574T.

```

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

```

Kode `Serial.begin` adalah perintah untuk membuka port data serial untuk komunikasi serial. Jika tidak ingin menggunakan serial monitor kode tersebut dapat dihapus saja. Kemudian kode

`lcd.init` adalah inisialisasi untuk LCD yang kita gunakan, apabila tidak kita tuliskan maka ketika kita upload program tidak akan ada respon apapun. Letakan kode `lcd.init` tersebut pada `void setup`, hal itu dilakukan agar LCD tidak berkedip terus menerus ketika menampilkan data.

```
lcd.backlight();
```

`lcd.backlight` berguna untuk mengaktifkan cahaya pada LCD. Apabila tidak kita berikan kode tersebut, maka LCD akan gelap walaupun ada data yang ditampilkan.

```
lcd.setCursor(0,0);  
lcd.print("Hello");  
lcd.setCursor(0,1);  
lcd.print("LCD I2C TEST");
```

Kode selanjutnya menunjukkan perintah untuk memulai kursor/tulisan pada titik awal LCD yaitu pojok kiri atas, dengan memunculkan tulisan "Hello". Sedangkan untuk kode berikutnya tulisan diletakkan di baris kedua dengan tampilan "LCD I2C TEST".

Project 6 - Sensor Jarak (HC-SR04) dengan Serial Monitor

8.1 Sensor Ultrasonik

Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat di dengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik bisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa.

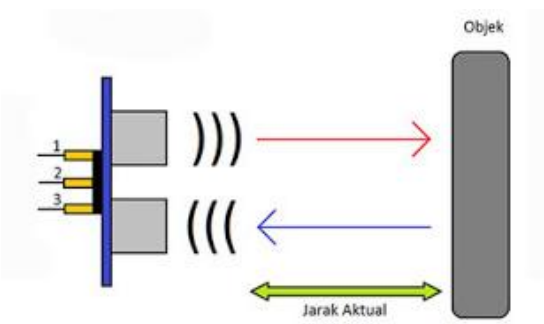
Sensor jarak yang akan kita buat tersusun oleh sensor ultrasonik HC-SR04 yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Sensor ini merupakan sensor ultrasonik siap pakai yang sudah terdiri dari pengirim, penerima, dan pengontrol gelombang ultrasonik. Alat ini bisa digunakan untuk mengukur jarak benda dari 2cm – 4m dengan

akurasi 3mm. Alat ini memiliki 4 pin, pin VCC, GND, Trigger, dan Echo. Pin VCC untuk listrik positif dan GND untuk ground-nya. Pin Trigger untuk trigger keluarnya sinyal dari sensor dan pin Echo untuk menangkap sinyal pantul dari benda.



Gambar 8.1 Sensor Ultrasonik HC-SR04

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Gelombang yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s.



Gambar 8.2 Cara kerja sensor ultrasonik

Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima. Ketika kita ketahui selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima, maka dapat dihitung jarak benda berdasarkan rumus berikut.

$$S = \frac{340 \left(\frac{m}{s} \right) t(s)}{2}$$

dimana S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh transmitter dan waktu ketika gelombang pantul diterima receiver. Agar nilai tersebut dapat terbaca oleh Arduino maka rentang waktu kerjanya harus kita ubah dalam satuan mikrosekon (μs) dan satuan jarak bisa kita ubah dalam satuan cm.

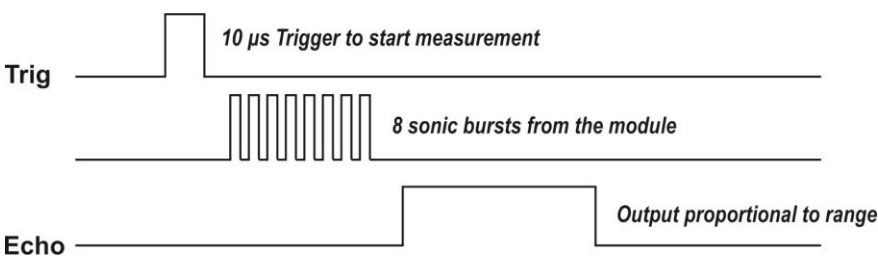
$$S = \frac{340 \frac{100 \text{ cm}}{1000000 \mu s} t(s)}{2} = \frac{0,034 \frac{cm}{\mu s} t(s)}{2}$$

atau

$$S = \frac{t(s)/2}{29,4117647 \mu s/cm}$$

Cara menggunakan alat ini yaitu ketika kita memberikan tegangan positif pada pin Trigger selama 10uS, maka sensor akan mengirimkan 8 step sinyal ultrasonik dengan frekuensi 40kHz. Selanjutnya, sinyal akan diterima pada pin Echo. Untuk mengukur jarak benda yang memantulkan sinyal tersebut, maka selisih waktu

ketika mengirim dan menerima sinyal digunakan untuk menentukan jarak benda tersebut. Berikut adalah visualisasi dari sinyal yang dikirimkan oleh sensor HC-SR04.



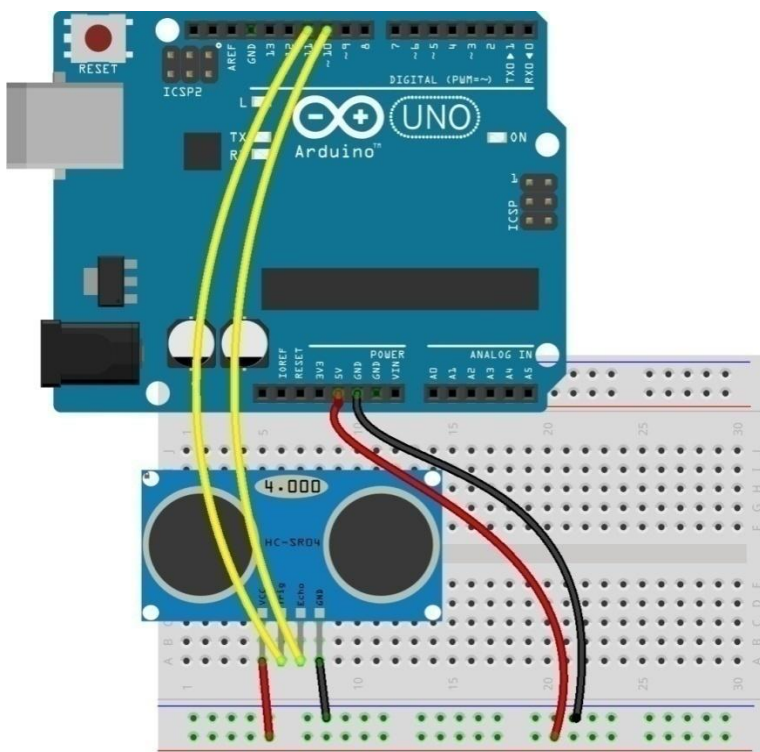
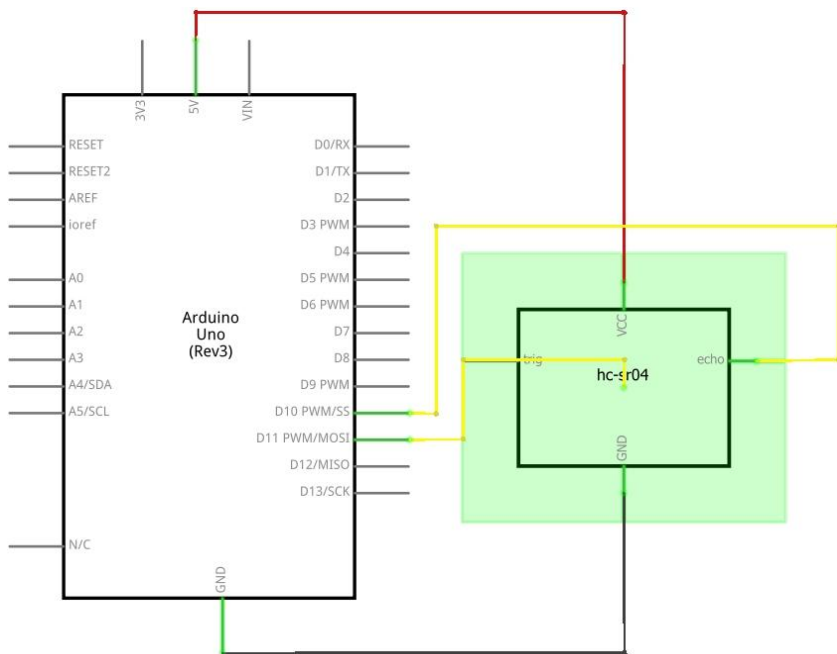
Gambar 8.3 Sistem pewaktu pada sensor HC-SR04

8.2 Rangkaian Sensor Jarak

Sensor jarak yang akan kita buat terdiri dari beberapa komponen diantaranya yaitu,

Komponen	Jumlah
Sensor Ultrasonik HC-SR04	1
papan rangkaian	1
Arduino UNO	1
Kabel Jumper	6

Untuk lebih jelasnya perhatikan rangkaian dari sensor ultrasonik pada **Gambar 8.4** berikut ini.



Gambar 8.4 Rangkaian Sensor Jarak

- Pasangkan kabel jumper dari pin GND dan 5 V Arduino ke bagian bawah papan rangkaian
- Pasangkan Sensor Ultrasonik HC-SR04 ke papan rangkaian dimana VCC ke 5 V pada papan rangkaian, Trig ke Pin 10 Arduino, Echo ke Pin 11 Arduino, dan GND ke GND papan rangkaian
- Untuk pemasangan kabel jumper, bisa dengan warna yang berbeda atau lubang yang berbeda, asalkan komponen dan kabel penghubung sesuai dengan gambar diatas.

8.3 Program Sensor Jarak

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
//Inisialisasi
#define trigPin 11
#define echoPin 10

void setup(){
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop(){
  int duration, distance;
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2)/29.1;

  Serial.print("Jarak Objek = ");
  Serial.println(distance);
}
```

```
Serial.print(" cm");  
delay(1000);  
}
```

Sensor ultrasonik HC-SR04 memiliki 4 pin yaitu VCC, *Trigger*, *Echo*, dan GND. Pin *Trigger* diletakkan pada pin 12 Arduino sedangkan pin *Echo* diletakkan pada pin 11 Arduino. *Trigger* berfungsi sebagai *output* atau penghasil sinyal ultrasonik dan *Echo* berfungsi sebagai *input* atau penerima sinyal.

```
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);
```

Saat `trigPin` bernilai HIGH, maka sensor akan mengeluarkan sinyal ultrasonik selama 10 μ s, dilanjutkan dengan `trigPin` bernilai LOW maka sensor akan berhenti mengeluarkan sinyal ultrasonik.

```
duration = pulseIn(echoPin, HIGH);  
distance = (duration/2)/29.1;
```

Bersamaan dengan itu, fungsi `pulseIn` akan memerintahkan sistem untuk menunggu hingga pin *Echo* bernilai HIGH. Lamanya proses menunggu akan dianggap sebagai durasi pengiriman + penerimaan sinyal *Echo* yang dipantulkan oleh benda. Durasi tersebut akan digunakan untuk menghitung jarak dari benda tersebut dalam satuan cm.

Project 7 - Sensor Jarak (HC-SR04) dengan Modul LCD 16x2

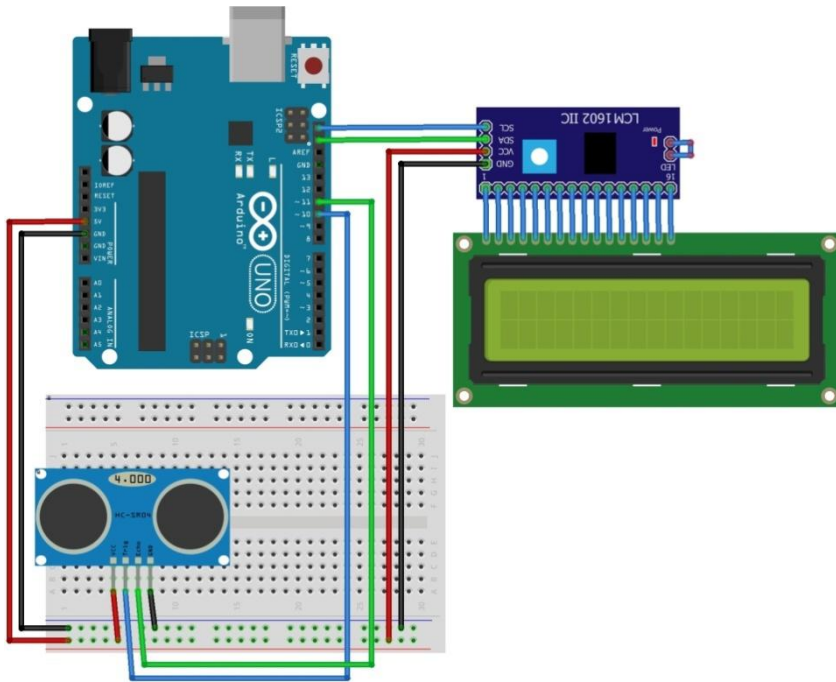
Pada *project* ini akan melanjutkan bahasan pada *project* sebelumnya. Jika pada bahasan sebelumnya hasil pengukuran ditampilkan di Serial Monitor IDE Arduino, maka pada bahasan ini akan ditampilkan menggunakan modul LCD 16x2 yang sudah dilengkapi dengan I2C Konverter.

9.1 Rangkaian

Sensor jarak yang akan kita buat terdiri dari beberapa komponen diantaranya yaitu,

Komponen	Jumlah
Sensor Ultrasonik HC-SR04	1
Arduino UNO	1
Kabel Jumper	10
LCD 16x2 dengan modul I2C Konverter	1
papan rangkaian	1

Untuk lebih jelasnya perhatikan rangkaian dari sensor ultrasonik berikut.



Gambar 9.1 Rangkaian Sensor Jarak (HC-SR04) dengan LCD 16x2 dan I2C Konverter

Buatlah rangkaian seperti pada **Gambar 9.1**.

- Pin GND dan 5 V Arduino ke papan rangkaian
- LCD 16x2 dengan modul I2C Konverter, dimana VCC ke 5 V papan rangkaian, GND ke GND papan rangkaian. SDA ke SDA Arduino dan SCL ke SCL Arduino.
- Sensor Ultrasonik HC-SR04, dimana VCC ke VCC papan rangkaian, Trig ke Pin 10 Arduino, Echo ke Pin 11 Arduino, dan GND ke GND papan rangkaian.

- Untuk pemasangan kabel jumper, bisa dengan warna yang berbeda atau lubang yang berbeda, asalkan komponen dan kabel penghubung sesuai dengan gambar diatas.

9.2 Program

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
float trigPin = 11;
float echoPin = 10;

LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
    Serial.begin (9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    lcd.init();
}

void loop()
{
    lcd.backlight();

    int duration, distance;
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2)/29,4117647;

    lcd.setCursor(0,0);
    lcd.print("Jarak Benda = ");
    lcd.setCursor(0,1);
    lcd.print("= ");
```



```

    lcd.print(distance);
    lcd.print("cm");
    delay(1000);
}

```

Untuk program yang lebih singkat, kita dapat menambahkan *Library* khusus untuk modul ultrasonic HC-SR04 dengan *keyword* **HCSR04** by **Martin Sasic**. Setelah kita tambahkan, dapat kita gunakan program seperti berikut ini.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <HCSR04.h>

LiquidCrystal_I2C lcd(0x27,16,2);
UltraSonicDistanceSensor    distanceSensor(11,
    10);

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

void loop()
{
    lcd.backlight();

    lcd.setCursor(0,0);
    lcd.print("Jarak Benda = ");
    lcd.setCursor(0,1);
    lcd.print("=" );
    lcd.print(distanceSensor.measureDistanceCm());
    lcd.print(" cm");
    delay(1500);
}

```

Program diatas diawali dengan inisialisasi *library* dari LCD 16x2 yang disertai modul I2C Konverter dan sensor ultrasonik HC-SR04.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <HCSR04.h>
```

Dilanjutkan dengan menuliskan alamat dari modul I2C dan inisialisasi pin sensor ultrasonik HC-SR04.

```
LiquidCrystal_I2C lcd(0x27,16,2);
UltraSonicDistanceSensor distanceSensor(11, 10);
```

Kode 0x27 diatas adalah alamat untuk modul I2C dengan tipe IC PCF8574T. Kemudian kode `UltraSonicDistanceSensor distanceSensor` adalah kode *default* yang berasal dari *library* sensor ultrasonik HC-SR04, sehingga kita harus menggunakannya ketika menggunakan *library* tersebut. `trigPin` diletakkan pada pin 11 dan `echoPin` pada pin 10.

```
void setup()
{
  Serial.begin (9600);
  lcd.init();
}
```

Kode `Serial.begin` adalah perintah untuk membuka port data serial untuk komunikasi serial. Jika tidak ingin menggunakan serial monitor kode tersebut dapat dihapus saja. Kemudian kode `lcd.init` adalah inisialisasi untuk LCD yang kita gunakan, apabila tidak kita tuliskan maka ketika kita upload program tidak akan ada respon apapun.

```
lcd.backlight();
```

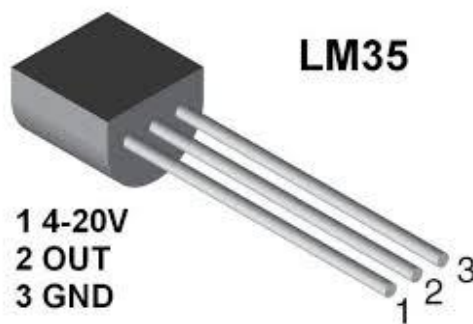
`lcd.backlight` berguna untuk mengaktifkan cahaya pada LCD. Apabila tidak kita berikan kode tersebut, maka LCD akan gelap walaupun ada data yang ditampilkan.

```
lcd.setCursor(0,0);  
lcd.print("Jarak Benda = ");  
lcd.setCursor(0,1);  
lcd.print("=" );  
lcd.print(distanceSensor.measureDistanceCm());  
lcd.print(" cm");  
delay(1500);
```

Kode selanjutnya menunjukkan perintah untuk menampilkan hasil pembacaan sensor di LCD. Dimulai dari baris 1 yang memunculkan "Jarak Benda = " kemudian pada baris ke-2 menampilkan hasil pembacaan sinyal digital dari sensor dalam satuan jarak “cm”.

10.1 Sensor Suhu LM35

Pada proses deteksi suhu, digunakan suatu komponen sensor yang dapat merubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Salah satu sensor suhu yang sering digunakan adalah sensor Suhu LM35, yang diproduksi oleh National Semiconductor. LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.



Gambar 10.1 Sensor Suhu LM35

Walaupun tegangan maksimum LM35 dapat mencapai 30 V, tetapi rekomendasi tegangan yang digunakan adalah 5 V, sehingga tidak memerlukan catu daya dari luar cukup dengan menghubungkannya ke Arduino. Sensor LM35 juga tidak membutuhkan kalibrasi, dikarenakan sampai lebih kurang seperempat derajat celcius pada temperature ruang.

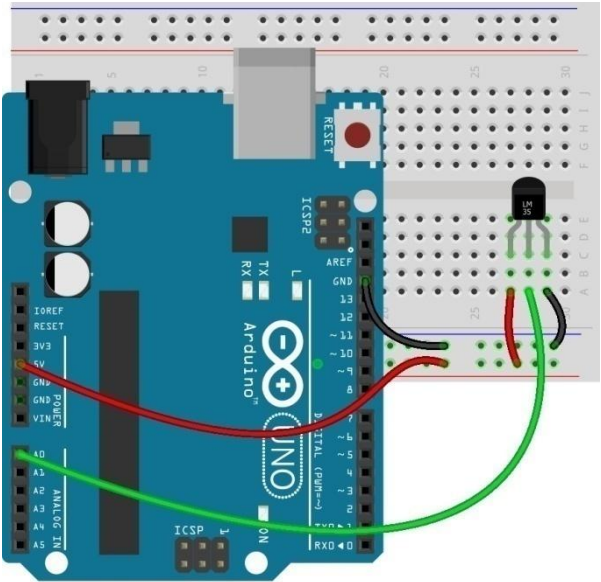
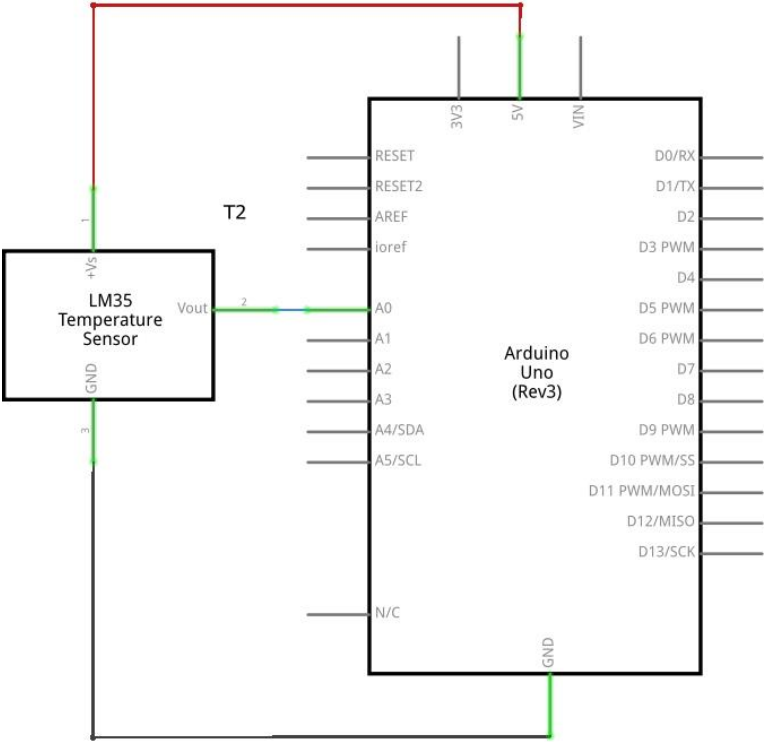
Untuk membaca data sensor ini, kita perlu merubah data analog berupa tegangan menjadi data digital pada Arduino. LM35 merespon perubahan suhu sebesar 10mV/derajat celcius, untuk itu kita perlu merubah nilai tegangan yang dihasilkan menjadi milivolt. Dengan menggunakan Arduino yang menggunakan ADC 10bit atau sama dengan 1024 dan tegangan referensi yang digunakan yaitu 5 V, maka didapatkan persamaan berikut ini

$$\begin{aligned} \text{Derajat Celcius} &= \frac{ADC}{1024} \frac{5000\text{ mV}}{10\text{ mV}} \\ \text{Derajat Celcius} &= ADC \times 0,48828125 \end{aligned}$$

Pada *project* kali ini kita akan mencoba membuat rangkaian sensor suhu menggunakan sensor suhu LM35N. Berikut ini komponen-komponen yang dibutuhkan.

Komponen	Jumlah
Arduino Uno	1
Papan Rangkaian	1
Sensor Suhu LM35N	1
Kabel penghubung	5

10.2 Rangkaian Sensor Suhu



Gambar 10.2 Rangkaian sensor suhu

10.3 Program Sensor Suhu

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
//Inisialisasi pin
float temp;
int tempPin = A0;

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

void loop()
{
  temp = analogRead(tempPin);
  temp = temp * 0.48828125;
  Serial.print("TEMPRATURE = ");
  Serial.print(temp);
  Serial.print("*C");
  Serial.println();
  delay(1000);
}
```

Pada program tersebut, temp digunakan sebagai inisialisasi pembacaan suhu secara analog, dimana pin analog input kita inisialisasikan sebagai tempPin di pin A0.

```
float temp;
int tempPin = A0;
```

pembacaan data analog dengan inisialisasi temp

```
temp = analogRead(tempPin);
```

Kemudian nilai tersebut akan diubah dalam satuan temperatur dengan cara merubah data analog yang dibaca menjadi data digital.

```
temp = temp * 0.48828125;
```

kode diatas merupakan persamaan yang digunakan untuk mengkonversi data analog ke data digital dan sudah dalam satuan temperatur celcius. Selanjutnya data tersebut akan ditampilkan dalam serial monitor, untuk itu kita harus menambahkan perintah untuk membuka port data serial untuk komunikasi serial.

```
Serial.begin(9600);
```

Jika tidak menggunakan serial monitor tidak harus menggunakan kode tersebut.

```
Serial.print("TEMPRATURE = ");  
Serial.print(temp);  
Serial.print("*C");  
Serial.println();
```

Kode selanjutnya menunjukkan perintah untuk menampilkan hasil pembacaan sensor di serial monitor. Diawali dengan memunculkan "TEMPRATURE = " dilanjutkan dengan menampilkan hasil pembacaan sinyal digital dari sensor dalam satuan celcius.

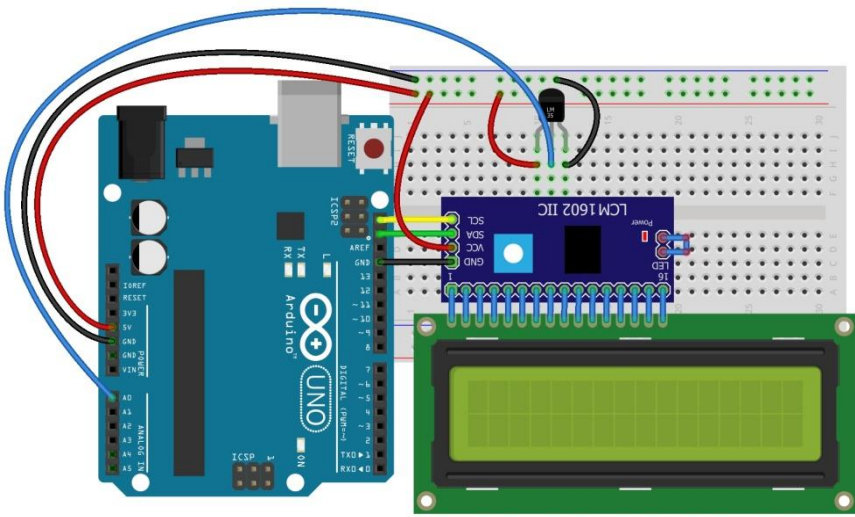
Project 9 - Sensor Suhu (LM35) dengan Modul LCD 16x2

Pada *project* ini akan melanjutkan bahasan pada *project* sebelumnya. Jika pada bahasan sebelumnya hasil pengukuran sensor suhu ditampilkan di Serial Monitor IDE Arduino, maka pada bahasan ini akan ditampilkan menggunakan LCD 16x2 yang sudah dilengkapi modul I2C Konverter.

11.1 Rangkaian

Berikut ini adalah komponen yang dibutuhkan untuk membuat *project* ini.

Komponen	Jumlah
Arduino Uno	1
Papan Rangkaian	1
Sensor Suhu LM35N	1
Kabel penghubung	9
LCD 16x2 dengan modul I2C Konverter	1



Gambar 11.1 Rangkaian Sensor Suhu (LM35) dengan Modul LCD

Buatlah rangkaian seperti pada **Gambar 11.1**.

- Pin GND dan 5 V Arduino ke papan rangkaian
- LCD 16x2 dengan modul I2C Konverter, dimana VCC ke 5 V papan rangkaian, GND ke board Arduino. SDA ke SDA Arduino dan SCL ke SCL Arduino.

Catatan:

Untuk pemasangan Modul I2C Konverter, pin GND diletakkan berbeda dari pin GND sensor, karena dapat mempengaruhi hasil pengukuran.

- Sensor LM35, dimana VCC ke VCC papan rangkaian, Output ke pin A0, dan GND ke GND papan rangkaian.
- Untuk pemasangan kabel jumper, bisa dengan warna yang berbeda atau lubang yang berbeda, asalkan komponen dan kabel penghubung sesuai dengan gambar diatas.

11.2 Program

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
float temp;
int tempPin = A0;

LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
    Serial.begin (9600);
    pinMode(tempPin, INPUT);
    lcd.init();
}

void loop()
{
    temp = analogRead(tempPin);
    temp = temp * 0.48828125;

    lcd.backlight();

    lcd.setCursor(0,0);
    lcd.print("TEMPRATURE");
    lcd.setCursor(0,1);
    lcd.print("=" );
    lcd.print(temp);
    lcd.print("*C");
    delay(1000);
}
```

Program diatas hampir sama dengan program pada project Sensor Suhu (LM35). Program diatas diawali dengan inisialisasi *library* dari LCD 16x2 yang disertai modul I2C Konverter dan inisialisasi

sensor suhu secara analog, dimana pin analog input kita inisialisasikan sebagai tempPin di pin A0.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
float temp;
int tempPin = A0;
```

Dilanjutkan dengan menuliskan alamat dari modul I2C.

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

Pada void setup, Serial.begin berfungsi untuk membuka port data serial untuk komunikasi serial baik mengirim atau menerima data dari serial. Jika tidak ingin menggunakan serial monitor kode tersebut dapat dihapus saja.

```
Serial.begin (9600);
pinMode(tempPin, INPUT);
lcd.init();
```

pinMode digunakan untuk menginisialisasikan tempPin sebagai *input* analog dengan resolusi 10 bit. Kemudian kode lcd.init adalah inisialisasi untuk LCD yang kita gunakan, apabila tidak kita tuliskan maka ketika kita upload program tidak akan ada respon apapun.

Pembacaan data analog dengan inisialisasi temp.

```
temp = analogRead(tempPin);
```

Kemudian nilai tersebut akan diubah dalam satuan temperatur dengan cara merubah data analog yang dibaca menjadi data digital.

```
temp = temp * 0.48828125;
```

kode diatas merupakan persamaan yang digunakan untuk mengkonversi data analog ke data digital dan sudah dalam satuan temperatur celcius. Selanjutnya data tersebut akan ditampilkan di LCD.

```
lcd.backlight();
```

`lcd.backlight` berguna untuk mengaktifkan cahaya pada LCD. Apabila tidak kita berikan kode tersebut, maka LCD akan gelap walaupun ada data yang ditampilkan.

```
lcd.setCursor(0,0);  
lcd.print("TEMPRATURE");  
lcd.setCursor(0,1);  
lcd.print("=" );  
lcd.print(temp);  
lcd.print("*C");  
delay(1000);
```

Kode selanjutnya menunjukkan perintah untuk menampilkan hasil pembacaan sensor di LCD. Dimulai dari baris 1 yang memunculkan "TEMPRATURE = " kemudian pada baris ke-2 menampilkan hasil pembacaan sinyal digital dari sensor dalam satuan celcius.

Project 10 - Kontrol Motor Stepper

12.1 Motor Stepper

Motor stepper adalah salah satu jenis motor yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Gerakan motor stepper berdasarkan urutan pulsa yang masuk ke motor, oleh karena itu untuk menggerakkan motor stepper memerlukan pengendali motor yang membangkitkan pulsa-pulsa periodik. Motor stepper memiliki keunggulan sebagai berikut:

1. Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur.
2. Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
3. Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
4. Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran)
5. Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya
6. Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada range yang luas.

Prinsip kerja motor stepper mirip dengan motor DC, dimana untuk menghasilkan medan magnet pada motor perlu adanya tegangan masuk. Motor DC memiliki magnet tetap pada stator sedangkan motor stepper memiliki motor tetap pada rotor. Motor stepper bekerja dengan mengkonversi bit-bit masukan menjadi gerakan rotor. Bit-bit tersebut berasal dari terminal-terminal input yang ada pada motor stepper yang menjadi kutub-kutub magnet dalam motor.



Gambar 12.1 Motor stepper dan kontroler

Apabila terminal diberi tegangan, terminal tersebut akan mengaktifkan kutub di dalam magnet sebagai kutub utara dan kutub yang tidak diberi tegangan sebagai kutub selatan. Dengan terdapatnya dua kutub di dalam motor ini, rotor di dalam motor yang memiliki kutub magnet permanen akan mengarah sesuai dengan kutub-kutub input. Kutub utara rotor akan mengarah ke kutub selatan stator sedangkan kutub selatan rotor akan mengarah ke kutub utara stator.

Motor stepper tidak dapat bergerak sendiri secara kontinyu, tetapi bergerak secara diskrit dari satu step ke step berikutnya. Motor stepper menghasilkan torsi yang besar dengan kecepatan rendah. Tiap-tiap motor stepper mempunyai spesifikasi masing-masing, antara lain 0.72° per step, 1.8° per step, 3.6° per step, 7.5° per step, 15° per step, dan bahkan ada yang 90° per step. Dalam pengoperasiannya kita dapat menggunakan 2 prinsip yaitu full step atau half step. Dengan full step berarti motor stepper berputar sesuai dengan spesifikasi derajat per stepnya, sedangkan half step berarti motor stepper berputar setengah derajat per step dari spesifikasi motor stepper tersebut.

Berdasarkan struktur rotor dan stator pada motor stepper, maka motor stepper dapat dikategorikan dalam 3 jenis sebagai berikut :

- Motor Stepper Variable Reluctance (VR)

Motor ini terdiri atas sebuah rotor besi lunak dengan beberapa gerigi dan sebuah lilitan stator.

- Motor Stepper Permanent Magnet (PM)

Motor stepper jenis ini memiliki rotor yang berbentuk seperti kaleng bundar (tin can) yang terdiri atas lapisan magnet permanen yang diselang-seling dengan kutub yang berlawanan.

- Motor Stepper Hybrid (HB)

Motor stepper tipe hibrid memiliki struktur yang merupakan kombinasi dari kedua tipe motor stepper sebelumnya.

Berdasarkan metode perancangan rangkaian pengendalinya, motor stepper dapat dibagi menjadi 2 jenis yaitu;

- Motor Stepper Unipolar

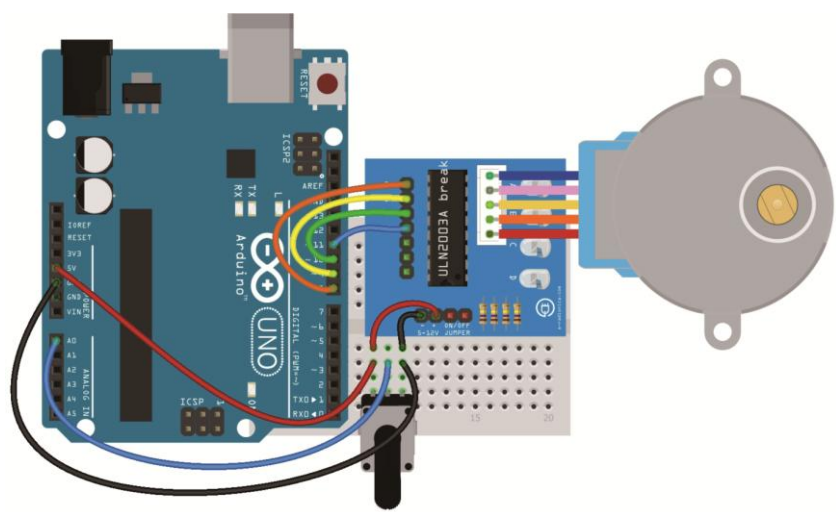
- Motor Stepper Bipolar

12.2 Rangkaian

Pada project ini akan dibahas mengenai kontrol motor stepper, dimana kita menggunakan kontroler dengan tipe IC ULN2003A. Berikut ini adalah komponen keseluruhan yang dibutuhkan.

Komponen	Jumlah
Arduino Uno	1
Kabel Jumper	9
Motor Stepper	1
Driver Motor Stepper ULN2003A	1
Potensiometer	1

Untuk lebih jelasnya perhatikan rangkaian dari kontrol motor stepper ini.



Gambar 12.2 Rangkaian kontrol motor stepper

Buatlah rangkaian seperti pada **Gambar 12.2**.

- Pin IN1 ke pin 11 Arduino
- Pin IN2 ke pin 10 Arduino
- Pin IN3 ke pin 9 Arduino
- Pin IN4 ke pin 8 Arduino
- 5-12V (+) ke VCC dan (–) ke GND
- Kaki 2 Potensiometer ke Pin A0 Arduino

12.3 Program

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
#include <Stepper.h>
const int stepsPerRevolution = 200;

Stepper myStepper(stepsPerRevolution, 8, 9, 10,
11);

void setup() {

}

void loop() {
    int sensorReading = analogRead(A0);
    int motorSpeed = map(sensorReading, 0, 1023, 0,
100);
    if (motorSpeed > 0) {
        myStepper.setSpeed(motorSpeed);
        myStepper.step(stepsPerRevolution / 100);
    }
}
```

Program diatas diawali dengan inisialisasi dari *library* yang digunakan yaitu **Stepper.h**, diikuti dengan inisialisasi jumlah step yang digunakan untuk sekali putaran.

```
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
```

Kode diatas juga merupakan inisialisasi dari *Library* stepper yang diletakkan pada pin 8, 9, 10, dan 11 Arduino.

```
int sensorReading = analogRead(A0);  
int motorSpeed = map(sensorReading, 0, 1023, 0, 100);
```

sensorReading digunakan untuk pembacaan data analog dari potensiometer. Kemudian data analog tersebut akan diubah dalam rentang 0-100 pada motorSpeed.

```
if (motorSpeed > 0) {  
    myStepper.setSpeed(motorSpeed);  
    myStepper.step(stepsPerRevolution / 100);  
}
```

Setelah hasil pembacaan potensiometer diubah dalam rentang 0-100, selanjutnya akan digunakan untuk memberikan input ke motor stepper sehingga kecepatan dari motor stepper dapat diatur menggunakan potensiometer.

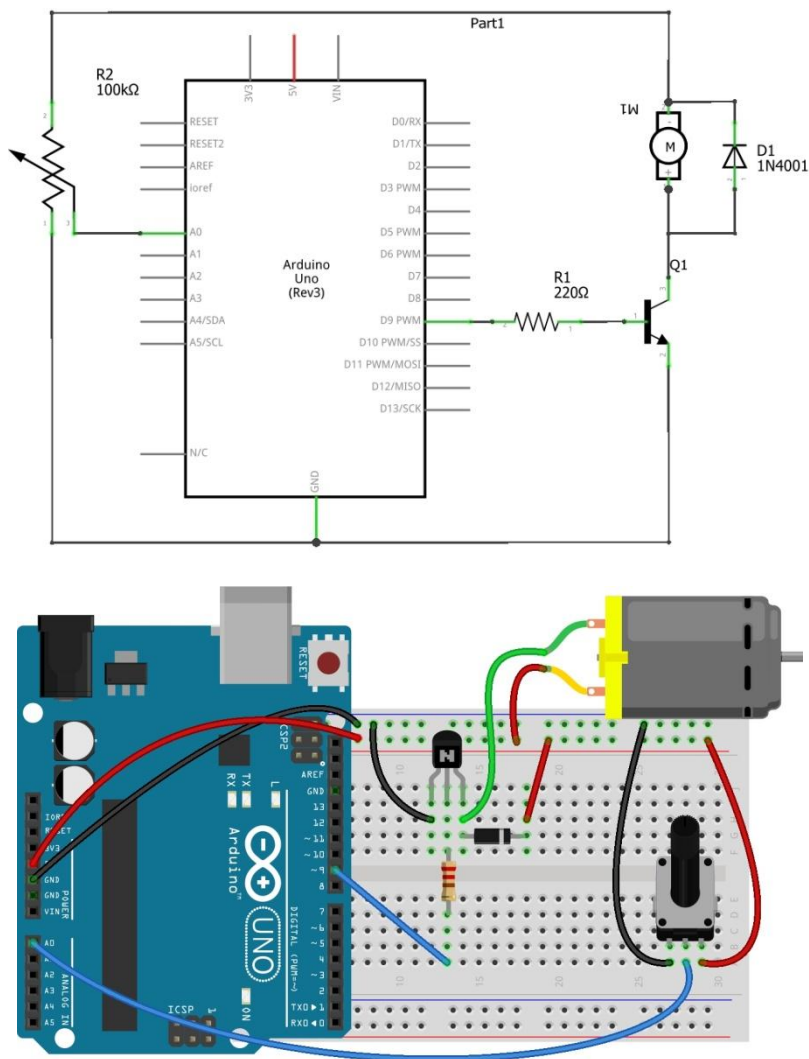
Project 11 - Kontrol Motor DC

Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/direct-unidirectional. Motor DC memiliki 3 bagian atau komponen utama untuk dapat berputar yaitu kutub magnet, dinamo dan commutator. Pada project ini akan dijelaskan tentang kontrol motor DC dengan menggunakan transistor TIP122. Sebelum membuat project ini, ada beberapa komponen yang harus dipersiapkan.

Komponen	Jumlah
Arduino Uno	1
Kabel Penghubung	10
Papan Rangkaian	1
Resistor 330 Ω	1
Dioda 1N4001/007	1
Transistor TIP122	1
Potensiometer	1
Motor DC 5V	1

13.1 Rangkaian

Buatlah rangkaian seperti pada **Gambar 13.1**.



Gambar 13.1 Rangkaian kontrol motor DC dengan PWM

13.1 Program

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno.

```
float potPin = 0;
float transistorPin = 9;
float potValue = 0;

void setup() {
  pinMode(transistorPin, OUTPUT);
}
void loop() {
  potValue = analogRead(potPin);
  analogWrite(transistorPin, potValue);
}
```

Program tersebut dimulai dengan inisialisasi variabel.

```
float potPin = 0;
float transistorPin = 9;
float potValue = 0;
```

`potPin` atau input analog dari potensiometer diletakkan di pin 0 Arduino, sedangkan `transistorPin` atau pin output ke transistor diletakkan di pin 9, dan juga diatur dalam `pinMode` sebagai OUTPUT.

```
pinMode(transistorPin, OUTPUT);
```

```
potValue = analogRead(potPin);
analogWrite(transistorPin, potValue);
```

Fungsi *analogRead* pada *potValue* digunakan untuk membaca nilai input analog potensiometer dengan resolusi 10 bit. Karena beresolusi 10 bit maka hasil pembacaan nilai digital antara 0 sampai 1023, tetapi dalam hal ini nilai analog dari potensiometer dibagi dengan 4, untuk itu nilainya berkisar dari 0 sampai 255. Kemudian fungsi *analogWriter* berfungsi untuk memberi nilai analog PWM pada output di pin 9/pin transistor.

Project 12 - Alarm Kebakaran (IR Flame)

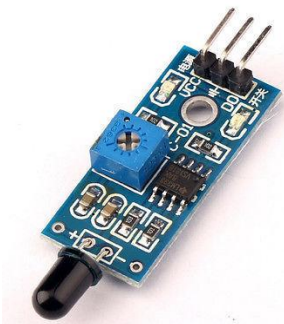
14.1 Sensor Api Infrared

Flame sensor atau sensor api merupakan sensor yang berfungsi untuk mendeteksi nyala api dengan menggunakan sensor infrared sebagai transduser dalam mendeteksi nyala apinya. Nyala api yang terbaca oleh sensor ini memiliki panjang gelombang antara 760 nm – 1100 nm. Suhu normal untuk pembacaan sensor ini yaitu 25 – 85 °C dengan sudut pembacaan 60°.

Prinsip kerja sensor ini adalah dengan mengidentifikasi nyala api dengan metode optik. Transduser yang digunakan dalam sensor ini berupa infrared (IR) yang mendeteksi cahaya pada panjang gelombang tertentu dan membedakan antara spektrum cahaya yang dipancarkan api dengan spektrum cahaya lainnya. Sensor api ini memiliki 3 pin kaki yang terdiri dari VCC, GND, dan DO.

Sensor api menggunakan data keluaran berupa data digital dari pin Output Digital (DO). Data digital dapat berupa logic LOW ataupun logic HIGH. Sensor ini dilengkapi dengan potensiometer

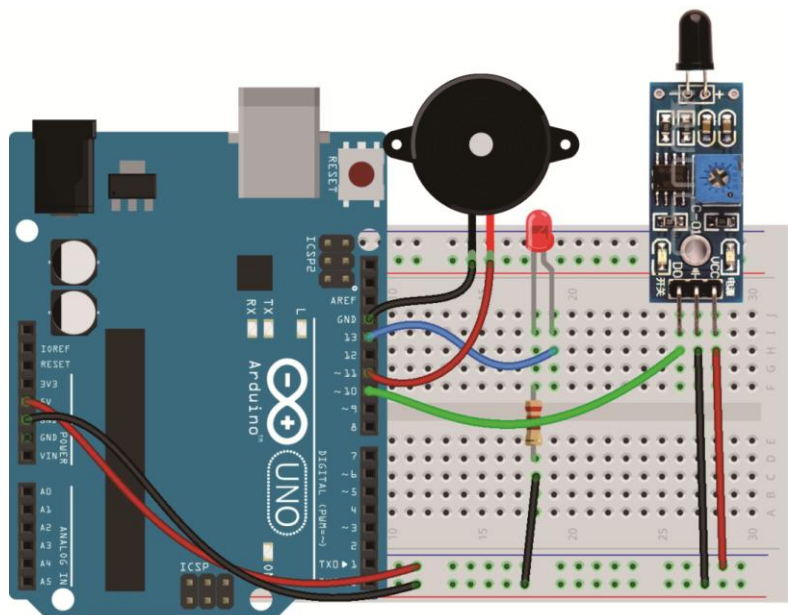
yang berfungsi untuk mengatur tingkat sensitifitas dari pembacaan sensor tersebut.



Gambar 14.1 Sensor api Infrared

14.2 Rangkaian

Pada project ini kita akan membuat sensor kebakaran dengan menggunakan sensor api infrared yang dilengkapi dengan alarm dari buzzer. Selanjutnya buatlah rangkaian seperti pada **Gambar 14.2**.



Gambar 14.2 Rangkaian alarm kebakaran

Berikut ini adalah komponen-komponen yang dibutuhkan untuk membuat project tersebut.

Komponen	Jumlah
Arduino Uno	1
Kabel Jumper	9
Sensor Api Infrared	1
Buzzer Aktif	1
LED Merah	1
Resistor 330 Ω	1

14.3 Program

Masukkan program berikut ini kedalam aplikasi IDE Arduino kemudian Upload ke Arduino Uno menggunakan kabel USB Downloader.

```
int pinApi = 10;
int pinAlarm = 11;
int redLED = 13;
int data;

void setup()
{
  pinMode(pinApi, INPUT);
  pinMode(pinAlarm, OUTPUT);
  pinMode(redLED, OUTPUT);
}
void loop()
{
  data = digitalRead(pinApi);
  if (data == LOW)
  {
    digitalWrite(pinAlarm, HIGH);
    digitalWrite(redLED, HIGH);
    delay(5000);
  }
}
```

```

else
{
    digitalWrite(redLED, LOW);
    digitalWrite(pinAlarm, LOW);
    delay(1000);
}
}

```

Sensor api yang kita gunakan menghasilkan data digital LOW dan HIGH. Pada kondisi tidak ada api yang terdeteksi, sensor akan menghasilkan data digital HIGH, dan sebaliknya ketika sensor mendeteksi api maka sensor akan mengeluarkan data digital LOW. Pada program diatas pin 10 Arduino kita jadikan sebagai Input dari sensor, sehingga Arduino akan membaca data digital dari sensor api yang digunakan.

```

pinMode(pinApi, INPUT);
pinMode(pinAlarm, OUTPUT);
pinMode(redLED, OUTPUT);

```

Data digital tersebut nantinya akan diproses untuk menghidupkan Alarm dan LED, sehingga pin Alarm dan pin LED kita atur sebagai Output. Ketika sensor bernilai LOW (mendeteksi api), pin Alarm dan LED kita atur dalam kondisi HIGH atau hidup.

```

data = digitalRead(pinApi);
if (data == LOW)
{
    digitalWrite(pinAlarm, HIGH);
    digitalWrite(redLED, HIGH);
    delay(5000);
}

```

Saat sensor bernilai HIGH (tidak mendeteksi api), pin Alarm dan LED kita atur dalam kondisi LOW (mati).

```

else
{

```

```
digitalWrite(redLED, LOW);  
digitalWrite(pinAlarm, LOW);  
delay(1000);  
}
```

Untuk pengembangannya dapat ditambahkan sensor asap/sensor gas metana agar lebih kompleks sehingga tidak hanya mendeteksi nyala apinya tetapi juga gas yang muncul ketika terjadi kebakaran.

DAFTAR PUSTAKA

Andrianto, H. dan Darmawan, A. 2016. Arduino Belajar Cepat dan Pemrograman. Informatika Bandung. Bandung.

McRoberts Mike. 2009. Arduino Starter Kit Manual. Earthshine Design.

Monk, Simon. 2010. 30 Arduino Projects for the Ecil Genius. McGraw-Hill Companies, Inc. United State.

Santoso, Hari. 2015. Panduan Praktis Arduino untuk Pemula. www.elangsakti.com. Trenggalek.

<http://www.belajArduino.com/2016/06/how-to-connect-1602-2004-iic-i2c-lcd-to.html>

<http://www.elangsakti.com>

<http://www.instructables.com>

<https://create.Arduino.cc>

<https://github.com>

<https://id.wikipedia.org/wiki/Arduino>

<https://www.arduino.cc>

<https://www.nyebarilmu.com/cara-mengakses-sensor-pir>

BIOGRAFI PENULIS



Dr. Junaidi, S.Si., M.Sc. dilahirkan di Suka Agung Kabupaten Tanggamus pada tanggal 18 Juni 1982. Menyelesaikan pendidikan S1 di Jurusan Fisika FMIPA Universitas Lampung pada tahun 2006. Pendidikan Magister (S2) dan Doktor (S3) diselesaikan di Jurusan Fisika UGM pada tahun 2012 dan 2016 pada bidang Instrumentasi Material.

Penulis tercatat sebagai dosen di Jurusan Fisika FMIPA Universitas Lampung sejak tahun 2008 sampai dengan sekarang. Saat ini penulis juga sebagai Kepala Laboratorium Elektronika Dasar Jurusan Fisika FMIPA Universitas Lampung.



Yuliyon Dwi Prabowo dilahirkan di Tulang Bawang 8 Juli 1997. Saat ini sedang menempuh pendidikan S1 di Jurusan Fisika FMIPA Universitas Lampung sejak tahun 2014, dan mengambil konsentrasi dalam bidang Instrumentasi. Pernah bersekolah di SD Negeri 1 Margodadi (2003-2009), SMP Negeri 1 Tumijajar (2009-2012), dan

melanjutkan ke SMA N 1 Terbanggi Besar di kelas Akselerasi (2012-2014). Penulis pernah aktif diorganisasi kemahasiswaan seperti Ketua Physics Instrumentation Club (PIC) bidang proyek dan aplikasi (2015-2016), Ketua Sains dan Teknologi Himpunan Mahasiswa Fisika FMIPA Unila (2016), dan Ketua Umum Physics Instrumentation Club (2017-2018).