

MODUL PRAKTIKUM MIKROKONTROLER



Arduino IDE 1.8.13

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

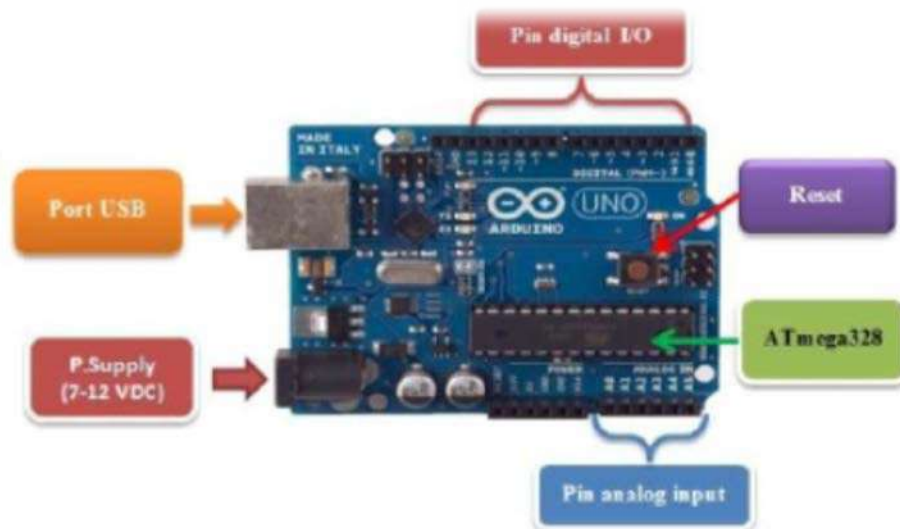
Windows Win 7 and newer
Windows ZIP file

Windows app Win 8.1 or 10 [Get](#)

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

Release Notes Checksums (sha512)



Oleh:
Farry Apriliano Haskari
Endo Argo Kuncoro

DAFTAR ISI

	Hal
HALAMAN JUDUL	1
DAFTAR ISI	2
BAB I PENDAHULUAN	
A. PENGANTAR TEKNOLOGI MIKROKONTROLER	3
B. MIKROKONTROLER PLATFORM ARDUINO	8
C. PEMROGRAMAN ARDUINO	16
BAB II PRAKTIKUM	
1 Percobaan LED Blink	22
2 Percobaan Push Button	23
3 Percobaan Loop	24
4 Percobaan Analog Input	26
5 Percobaan LCD Display	27
6 Percobaan Unipolar Motor Stepper	29
7 Percobaan Keypad	30
8 Percobaan Sensor Ultrasonik	32

BAB I

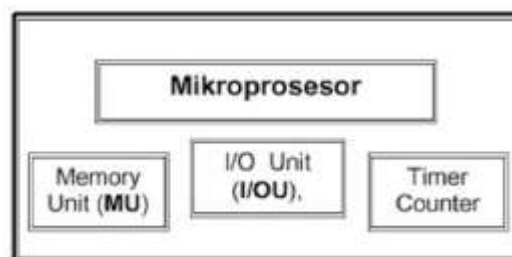
PENDAHULUAN

I. Pengantar Teknologi Mikrokontroler

1.1. Perkembangan Mikrokontroler

Sejarah awal perkembangan mikrokontroler dimulai pada awal tahun 1970-an. Motorola mengeluarkan seri mikrokontroler 6800 yang terus dikembangkan menjadi 68HC05, 68HC08, 68HC11, 68HC12, dan 68HC16. Zilog juga mengeluarkan seri mikroprosesor Z80-nya yang terkenal dan terus dikembangkan hingga kini menjadi Z180 dan kemudian diadopsi juga oleh mikroprosesor Rabbit. Tahun 1974, Texas Instrument mengenalkan mikrokontroler dengan seri TMS 1000 yang merupakan mikrokontroler 4 bit. TMS 1000 adalah mikrokomputer dalam sebuah chip, lengkap dengan RAM dan ROM. Tahun 1976, Intel mengeluarkan mikrokontroler 8-bit seri Intel 8748 yang merupakan mikrokontroler dari keluarga MCS 48. Intel juga mengeluarkan mikrokontroler-nya yang populer di dunia yaitu 8051, yang karena begitu populernya maka arsitektur 8051 tersebut kemudian diadopsi oleh vendor lain seperti Phillips, Siemens, Atmel, dan vendor-vendor lain. Selain itu masih ada mikrokontroler populer lainnya seperti Basic Stamps, PIC dari Microchip, MSP 430 dari Texas Instrument dan masih banyak lagi.

Mikrokontroler (*microcontroller*) atau disingkat dengan “*micron*” adalah pengendali yang merupakan suatu komputer kecil yang terletak di dalam sebuah chip atau IC (*integrated circuit*) yang berisikan inti prosesor, memori, dan komponen input/output yang dapat diprogram. Mikrokontroler biasa digunakan pada produk dan perangkat yang dapat dikontrol secara otomatis, seperti sistem kontrol mesin mobil (*engine control*), perangkat medis (*medical devices*), pengendali jarak jauh (*remote control*), mesin perkantoran (*office machines*), dan juga mainan (*games*). Penggunaan mikrokontroler lebih ekonomis dibandingkan sebuah desain sistem yang berisikan mikroprosesor, memori, dan perangkat input/ouput terpisah. Mikrokontroler adalah komputer mikro dalam satu chip tunggal. Mikrokontroler memadukan CPU, ROM, RWM, I/O paralel, I/O seri, counter-timer, dan rangkaian clock dalam satu chip tunggal seperti terlihat pada Gambar 1.1.



Gambar 1.1. Blok Diagram Mikrokontroler

Sama halnya dengan mikroprosesor, mikrokontroler adalah perangkat yang dirancang untuk kebutuhan umum (*specific purpose*). Sesuai dengan fungsinya sebagai pengendali, mikrokontroler berisikan sepaket chip lengkap yang terdiri dari fitur-fitur pengolah data yang juga terdapat dalam mikroprosesor, ditambah RAM, ROM, I/O, dan fitur lain yang terintegrasi di dalamnya. Contohnya dapat ditemui pada perangkat otomotif, mesin industri, elektronik dan perangkat- perangkat lain yang memiliki embedded sistem di dalamnya. Mikrokontroler sebagai sebuah chip telah mengalami perkembangan baik dari sisi arsitektur, teknologi dan kemampuannya. Untuk melihat perbedaan konsep diantara mikroprosesor dan mikrokontroler di bawah ini ditunjukkan Tabel 1.1 perbandingan konfigurasi, arsitektur, dan set instruksi diantara mikroprosesor Z-80 CPU dengan mikrokontroler AT89C51.

Tabel 1.1. Perbedaan Mikroprosesor dan Mikrokontroler

No	Konfigurasi Pin	Mikroprosesor Z-80 CPU	Mikrokontroler AT89C51
1	Jumlah pin	40	40
2	Jumlah pin alamat	16	16
3	Jumlah pin data	8	8
4	Jumlah pin interrupt	2	2
5	Pin I/O	-	32
6	Register 8 bit	20	34
7	Register 16 bit	4	2
8	Ukuran Stack	64 K	128
9	ROM Internal	-	4 Kbyte
10	RAM Internal	-	128 byte
11	Memori eksternal	64 Kbyte	128 Kbyte
12	Flag	6	4
13	Timer	0	2
14	Port Paralel	0	4 x 8 bit
15	Port Serial	0	1

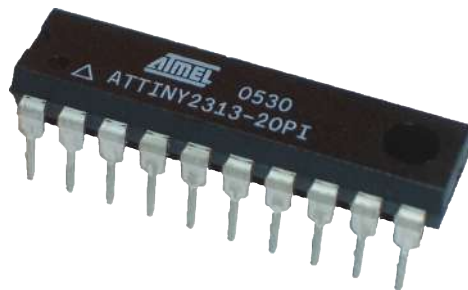
Terdapat perbedaan yang signifikan antara mikrokontroler dan mikroprocessor. Perbedaan yang utama antara keduanya dapat dilihat dari dua faktor utama yaitu arsitektur perangkat keras (*hardware architecture*) dan aplikasi masing-masing.

- 1) Ditinjau dari segi arsitekturnya, mikroprosesor hanya merupakan *single chip* CPU, sedangkan mikrokontroler dalam IC-nya selain CPU juga terdapat device lain yang memungkinkan mikrokontroler berfungsi sebagai suatu *single chip* komputer. Dalam sebuah IC mikrokontroler telah terdapat ROM, RAM, EPROM, serial interface dan paralel interface, timer, interrupt controller, konverter Analog ke Digital, dan lainnya (tergantung feature yang melengkapi mikrokontroler tersebut).
- 2) Sedangkan dari segi aplikasinya, mikroprocessor hanya berfungsi sebagai *Central Processing Unit* yang menjadi otak komputer, sedangkan mikrokontroler, dalam bentuknya yang mungil, pada umumnya ditujukan

untuk melakukan tugas-tugas yang berorientasi kendali (*control*) pada rangkaian yang membutuhkan jumlah komponen minimum dan biaya rendah (*low cost*).

1.2 Perkembangan Teknologi Mikrokontroler

Mikrokontroler sebagai teknologi baru yaitu teknologi semikonduktor kehadirannya sangat membantu perkembangan dunia elektronika. Dengan arsitektur yang praktis tetapi memuat banyak kandungan transistor yang terintegrasi, sehingga mendukung dibuatnya rangkaian elektronika yang lebih portable. Mikrokontroler dapat diproduksi secara masal sehingga harganya menjadi lebih murah dibandingkan dengan mikroprosesor, tetapi tetap memiliki kelebihan yang bisa diandalkan. Gambar 1.2 memperlihatkan beberapa contoh chip mikrokontroler.



Gambar 1.2. Chip Mikrokontroler

Mikrokontroler memiliki perbandingan ROM dan RAM-nya yang besar, artinya program kontrol disimpan dalam ROM (bisa *Masked ROM* atau *Flash PEROM*) yang ukurannya relatif lebih besar, sedangkan RAM digunakan sebagai tempat penyimpanan sementara, termasuk register-register yang digunakan pada mikrokontroler yang bersangkutan.

Perkembangan Mikrokontroler mengalami perubahan dari segi rancangan dan aplikasinya, seperti faktor kecepatan pengolah data yang semakin meningkat (cepat) dibanding pendahulunya. Seperti halnya sebuah mikroprosesor, mikrokontroler juga berkembang sesuai rancangan dan model-model aplikasinya. Mikrokontroler berdasarkan jumlah bit data yang dapat diolah dapat dibedakan dalam :

- 1) **Mikrokontroler 4 Bit**; merupakan mikrokontroler dengan jumlah bit data terkecil. Mikrokontroler jenis ini diproduksi untuk meminimalkan jumlah pin dan ukuran kemasan.
- 2) **Mikrokontroler 8 Bit**; merupakan mikrokontroler yang paling banyak digunakan untuk pekerjaan-pekerjaan perhitungan skala kecil. Dalam komunikasi data, Data ASCII serial juga disimpan dalam ukuran 8 bit. Kebanyakan IC memori dan fungsi logika dibangun menggunakan data 8 bit sehingga interface bus data

menjadi sangat mudah dibangun. Penggunaan mikrokontroler 8 bit jauh lebih banyak dibandingkan dengan mikrokontroler 4 bit. Aplikasinya juga sangat variatif mulai dari aplikasi kendali sederhana sampai kendali mesin berkecepatan tinggi.

- 3) **Mikrokontroler 16 Bit**; keterbatasan-keterbatasan yang ada pada mikrokontroler 8 bit berkaitan dengan semakin kompleksnya pengolahan data dan pengendalian serta kecepatan tanggap/respon disempurnakan dengan menggunakan mikrokontroler 16 bit. Salah satu solusinya adalah dengan menaikkan kecepatan clock, dan ukuran data. Mikrokontroler 16 bit digunakan untuk mengatur tangan robot, dan aplikasi *Digital Signal Processing* (DSP).
- 4) **Mikrokontroler 32 Bit**; ditargetkan untuk aplikasi Robot, Instrumen cerdas, Avionics, Image Processing, Telekomunikasi, Automobil, dan sebagainya. Program-program aplikasinya bekerja dengan sistem operasi dan dipadukan dengan perangkat pintar lainnya.

Karena kebutuhan yang tinggi terhadap “*smart chip*” dengan berbagai fasilitasnya, maka berbagai vendor juga berlomba untuk menawarkan produk-produk mikrokontrolernya. Selain mikroprosesor dan mikrokontroler, sebenarnya telah bermunculan chip-chip pintar lain seperti DSP prosesor dan *Application Specific Integrated Circuit* (ASIC). Di masa depan, chip-chip mungil berkemampuan sangat tinggi akan mendominasi semua desain elektronik di dunia sehingga mampu memberikan kemampuan komputasi yang tinggi serta meminimumkan jumlah komponen-komponen konvensional.

1.3. Implementasi Teknologi Mikrokontroler

Karena kemampuannya yang tinggi, bentuknya yang kecil, konsumsi dayanya yang rendah, dan harga yang murah maka mikrokontroler begitu banyak digunakan di dunia. Mikrokontroler digunakan mulai dari mainan anak-anak, perangkat elektronik rumah tangga, perangkat pendukung otomotif, peralatan industri, peralatan telekomunikasi, peralatan medis dan kedokteran, sampai dengan pengendali robot serta persenjataan militer. Terdapat beberapa keunggulan yang diharapkan dari alat-alat yang berbasis mikrokontroler (*microcontroller-based solutions*):

- 1) Keandalan tinggi (*high reliability*) dan kemudahan integrasi dengan komponen lain (*high degree of integration*)
- 2) Ukuran yang semakin dapat diperkecil (*reduced in size*)
- 3) Penggunaan komponen dipersedikit (*reduced component count*) yang juga akan menyebabkan biaya produksi dapat semakin ditekan (*lower manufacturing cost*)
- 4) Waktu pembuatan lebih singkat (*shorter development time*) sehingga lebih cepat pula dijual ke pasar sesuai kebutuhan (*shorter time to market*)
- 5) Konsumsi daya yang rendah (*lower power consumption*)

Penerapan teknologi di masyarakat akan memberikan banyak keuntungan. Mikrokontroler merupakan sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output. Dengan penggunaan mikrokontroler ini maka :

- 1) Sistem elektronik akan menjadi lebih ringkas,
- 2) Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi,
- 3) Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak,

Teknologi mikrokontroler dapat diterapkan di berbagai bidang, baik di industri maupun di masyarakat. Dengan menerapkan teknologi Mikrokontroler di masyarakat akan memberikan banyak keuntungan, contohnya :

- 1) Kunci Kombinasi Digital dengan Remote Kontrol Inframerah
- 2) Jam Wekker Digital Bertampilan M1632 LCD
- 3) Alat penyiram tanaman otomatis
- 4) Pemberi pakan ikan otomatis
- 5) Deteksi kebakaran menggunakan sensor suhu dan mikrokontroler
- 6) Pendeteksi kebocoran gas
- 7) Sistem keamanan mobil
- 8) Pintu Air Otomatis
- 9) Pengendali lampu rumah dengan mikrokontroler melalui SMS
- 10) Robot Cerdas

1.4. Ringkasan

1. Mikrokontroler (*microcontroller*) adalah pengendali yang merupakan suatu komputer kecil yang terletak di dalam sebuah chip atau IC (*integrated circuit*) yang berisikan inti prosesor, memori, dan komponen input/output yang dapat diprogram.
2. Ditinjau dari segi arsitekturnya, mikroprosesor hanya merupakan single chip CPU, sedangkan mikrokontroler dalam IC-nya selain CPU juga terdapat device lain yang memungkinkan mikrokontroler berfungsi sebagai suatu single chip computer.
3. Mikrokontroler memiliki perbandingan ROM dan RAM-nya yang besar, artinya program kontrol disimpan dalam ROM (bisa Masked ROM atau Flash PEROM) yang ukurannya relatif lebih besar, sedangkan RAM digunakan sebagai tempat penyimpanan sementara, termasuk register-register yang digunakan pada mikrokontroler yang bersangkutan.

2. MIKROKONTROLER PLATFORM ARDUINO

2.1. Konsep Arduino

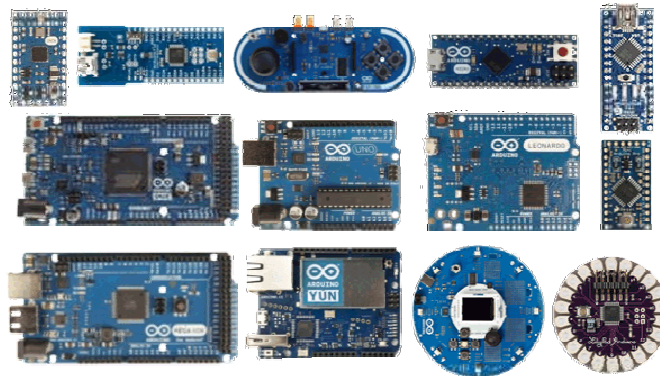
Arduino merupakan papan-tunggal mikrokontroler serba guna yang bisa diprogram dan bersifat *open-source*. Platform Arduino sekarang ini menjadi sangat populer dengan pertambahan jumlah pengguna baru yang terus meningkat. Hal ini karena kemudahannya dalam penggunaan dan penulisan kode program. Tidak seperti kebanyakan papan sirkuit pemrograman sebelumnya, Arduino tidak lagi membutuhkan perangkat keras terpisah (disebut *programmer* atau *downloader*) untuk memuat atau meng-*upload* kode baru ke dalam mikrokontroler. Cukup dengan menggunakan kabel USB untuk mulai menggunakan Arduino. Selain itu, Arduino IDE menggunakan bahasa pemrograman **C++** dengan versi yang telah disederhanakan, sehingga lebih mudah dalam belajar pemrograman. Arduino akhirnya berhasil menjadi papan sirkuit pemrograman paling disukai hingga menjadikannya sebagai bentuk standar dari fungsi mikrokontroler dengan paket yang mudah untuk diakses.

Hardware dan software Arduino dirancang bagi para seniman, desainer, pehobi, hacker, pemula dan siapapun yang tertarik untuk menciptakan objek interaktif dan pengembangan lingkungan. Arduino mampu berinteraksi dengan tombol, LED, motor, speaker, GPS, kamera, internet, ponsel pintar bahkan dengan televisi. Fleksibilitas ini dihasilkan dari kombinasi ketersediaan software Arduino yang gratis, papan perangkat keras yang murah, dan keduanya yang mudah untuk dipelajari. Hal inilah yang menciptakan jumlah pengguna menjadi sebuah komunitas besar dengan berbagai kontribusinya yang telah dirilis pada berbagai proyek dengan berbasiskan Arduino.

Arduino Uno merupakan versi terbaru dari keluarga Arduino, berbasis mikrokontroler ATmega328, menyempurnakan tipe sebelumnya, Duemilanove. Perbedaan Arduino tersebut adalah tidak menggunakan IC FTDI (*Future Technology Devices International*) *USB to Serial* sebagai driver komunikasi USB-nya tetapi menggunakan mikrokontroler ATmega8U2 yang diprogram sebagai konverter USB ke serial. Uno sendiri diambil dari bahasa Italia yang artinya 1 (satu). Gambar 2.1 adalah *board* Arduino Uno dengan spesifikasi hardware :

- Microcontroller : ATmega328
- Tegangan Operasi : 5 V
- Tegangan Input : 7 – 12V
- Digital I/O : 14 pin
- PWM : 6 channel
- Analog Input : 6 channel

- Memory : 32KB Flash PEROM (0,5 KB digunakan oleh *bootloader*), 2KB SRAM dan 1KB EEPROM
- Frekuensi Clock : 16 MH



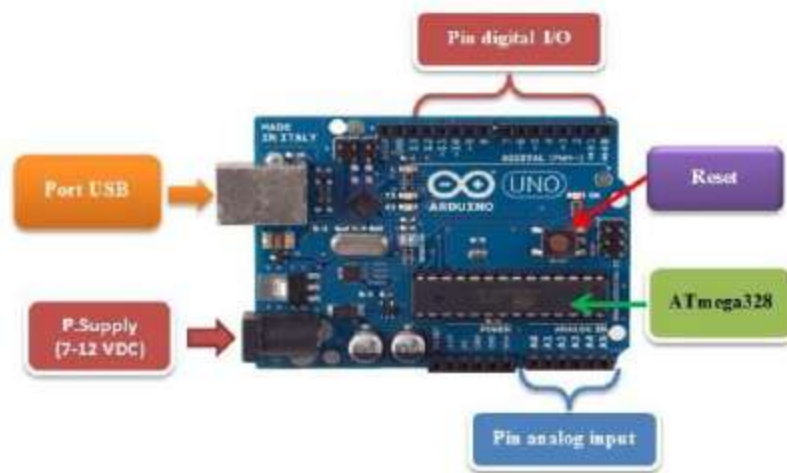
Gambar 2.1. Arduino Uno Board

Tabel 2.1. menunjukkan beberapa produk board yang diproduksi Arduino yang ada di pasaran :

Tabel 2.1 Produk Board Arduino

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

Gambar 2.2. memperlihatkan bagian utama dari papan Arduino uno, yakni terminal power supply, port USB, pin digital I/O, tombol reset, mikrokontroler ATmega328 dan pin analog input.



Gambar 2.2. Bagian Utama Arduino Uno Board

Berikut ini adalah penjelasan beberapa bagian utama dari papan Arduino uno, yaitu:

1. Power Supply.

Pada Arduino board, ada 2 (dua) pilihan sumber tegangan yang dapat digunakan, yakni dari port USB maupun dari power supply eksternal. Dengan menghubungkan port USB di komputer/laptop dengan Arduino maka secara otomatis *power supply* Arduino bersumber dari port USB. Untuk sumber tegangan eksternal (non-USB) yakni dengan menghubungkan Arduino board dengan sumber tegangan DC. Tegangan yang direkomendasikan adalah 7 sampai 12 V, jika kurang dari 7V akan menyebabkan ketidakstabilan tegangan, sedangkan jika lebih dari 12V akan menyebabkan panas dan akibat fatal berupa kerusakan pada board Arduino.

2. Input – Output.

Port Arduino berbeda penamaannya dengan sistem minimum mikrokontroler atau *microntroller development system*. Sebagai contoh pada system minimum ATmega8535 penamaan port adalah PORTA, PORTB, PORTC dan PORTD, untuk akses per-bit maka PORTA.0 s/d PORTA.7, contoh lain pada AT89S51 maka PORT0, PORT1 dan seterusnya. Sistem penamaan port pada Arduino merupakan urutan nomor port, mulai dari nomor nol (0), satu (1) dan seterusnya. Untuk digital I/O dengan nama pin 1, 2 sampai 13, sedangkan untuk analog input menggunakan nama A0, A1 sampai A5.

Pada Arduino uno terdapat 14 pin digital input – output. Secara umum berfungsi sebagai port input – output biasa, namun ada beberapa pin yang mempunyai fungsi alternatif. Sebagai contoh, pin 2 dan 3 juga sebagai pin interupsi eksternal. Kemudian pin 5,6,9,10 dan 11 dapat dipakai untuk PWM (*Pulse Width Modulation*) yang sering dipakai untuk kendali motor DC maupun motor servo. Tabel 4.2.menunjukkan nomor dan fungsi pin pada Arduino uno.

Tabel 2.2. Konfigurasi Pin Arduino Uno

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

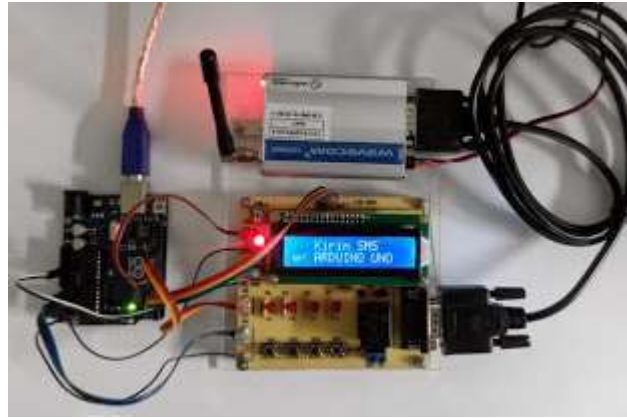
3. Analog Input.

Arduino memiliki 6 pin analog input, berfungsi membaca sinyal masukan analog seperti sensor analog. Meskipun demikian pin analog input dapat pula digunakan untuk keperluan digital I/O. Tabel 2.3. menunjukkan nomor dan fungsi pin input analog.

Tabel 2.3. Konfigurasi Pin Analog Input

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

Arduino memberikan kemudahan bagi penggunanya untuk membuat berbagai proyek berbasis mikrokontroler. Contohnya yang dapat dibuat dengan Arduino antara lain, untuk membuat simulasi lampu, membuat robot, mengontrol motor dc, mengontrol motor stepper, pengatur suhu, display LCD, dan masih banyak yang lainnya. Gambar 2.3 adalah contoh produk alat menggunakan Arduino.



Gambar 2.3 Kontrol Peralatan Listrik berbasis Arduino Uno

2.2. Perangkat Lunak Arduino

Arduino dapat digunakan untuk mengembangkan objek interaktif, mengambil masukan dari berbagai *switch* atau sensor, dan mengendalikan berbagai lampu, motor, dan output fisik lainnya. Proyek Arduino dapat berdiri sendiri, atau berkomunikasi dengan perangkat lunak (*software*) yang berjalan pada komputer (misalnya: Flash, Pengolahan, MaxMSP, database, dsb). Board dapat dirakit sendiri atau dibeli; open-source IDE dapat didownload secara gratis.

Arduino adalah software open source yang memudahkan untuk menulis kode program dan meng-upload-nya ke board Arduino. Software Arduino dapat berjalan pada Windows, Mac OS X, dan Linux. Software ini ditulis dalam bentuk Java dan berbasis processing, avr-gcc, dan perangkat lunak open source lainnya.

Software Arduino yang ada dalam situs Arduino (<https://www.arduino.cc/>) telah memiliki versi 1.6.6, seperti terlihat pada Gambar 2.4. Software Arduino IDE (*Integrated Development Environment*) adalah sebuah perangkat lunak yang memudahkan dalam mengembangkan aplikasi mikrokontroler mulai dari menuliskan source program, kompilasi, upload hasil kompilasi, dan uji coba secara terminal serial. Arduino dapat dijalankan di komputer dengan berbagai macam *platform* karena didukung atau berbasis Java. Source program yang dibuat untuk aplikasi mikrokontroler adalah bahasa C/C++ dan dapat digabungkan dengan *assembly*.



Gambar 2.4. Tampilan Software Arduino 1.8.13

Di samping IDE Arduino sebagai jantungnya, *bootloader* adalah jantung dari Arduino lainnya yang berupa program kecil yang dieksekusi sesaat setelah mikrokontroler diberi catu daya. *Bootloader* ini berfungsi sebagai pemonitor aktifitas yang diinginkan oleh Arduino. Jika dalam IDE terdapat file hasil kompilasi yang akan di-*upload*, *bootloader* secara otomatis menyambutnya untuk disimpan dalam memori program. Jika pada saat awal mikrokontroler bekerja, *bootloader* akan mengeksekusi program aplikasi yang telah diupload sebelumnya. Jika IDE hendak mengupload program baru, *bootloader* seketika menghentikan eksekusi program berganti menerima data program untuk selanjutnya diprogramkan dalam memori program mikrokontroler.

Hubungan komunikasi data antara IDE arduino dengan board Arduino digunakan komunikasi secara serial dengan protokol RS232. Jika board arduino sudah dilengkapi dengan komunikasi serial RS232 (biasanya USB), maka dapat langsung ditancapkan ke USB komputer. Piranti serial RS232 ini digunakan jika board arduino atau arduino buatan sendiri tidak dilengkapi dengan piranti serial 232.

Prosedur Menggunakan Arduino Board

1. Menyiapkan Arduino Board dan Kabel USB (Gambar 2.5)



Gambar 2.5. Arduino dan Kabel USB

2. Men-download Software Arduino

Dapatkan versi terbaru dari halaman download yang tersedia di situs resmi Arduino, <https://www.arduino.cc/en/Main/Software>. Setelah download selesai, unzip file yang didownload (jika mendownload file tipe .zip). Pastikan untuk mempertahankan struktur folder. Klik dua kali pada folder untuk membukanya, dan akan ada beberapa file dan sub-folder di dalam.

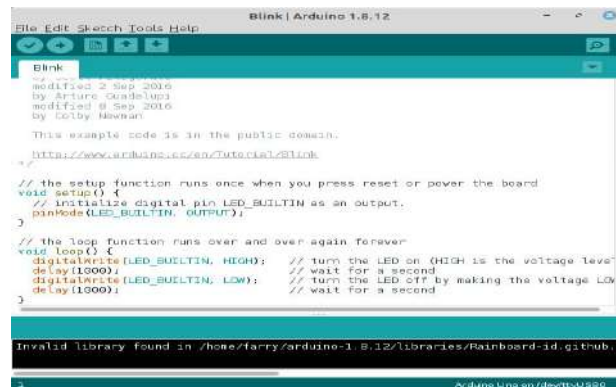
3. Menghubungkan Arduino Board ke Komputer

Arduino Uno, Mega Duemilanove dan Arduino Nano memerlukan sumber listrik dari salah satu koneksi USB komputer atau power supply eksternal. Sumber daya dipilih dengan jumper, plastik kecil yang terdapat antara USB dan jack listrik. Periksa apakah jumper diatur pada dua pin paling dekat dengan port USB atau tidak. Hubungkan papan Arduino ke komputer menggunakan kabel USB. LED indikator daya hijau (berlabel PWR) akan menyala.

4. Membuka Software Arduino (Gambar 2.6)

Klik dua kali aplikasi Arduino, lalu dapat dilakukan simulasi dengan hal-hal sebagai berikut :

- Buka contoh blink
- Buka LED berkedip contoh sketsa: File > Examples > 1.Basics > Blink.



```
File Edit Sketch Tools Help
Blink
// modified 2 Sep 2016
// by Arturo Guadalupe
// modified 8 Sep 2016
// by Colby Newman
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink

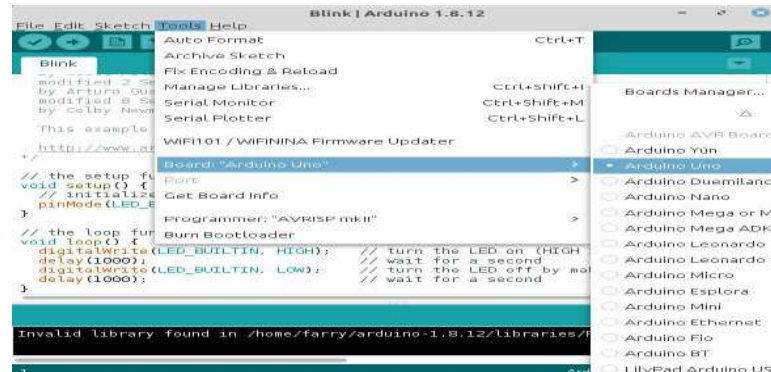
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Invalid library found in /home/ferry/arduino-1.8.12/libraries/Rainboard-1d.github.io
Arduino Uno on /dev/ttyUSB0
```

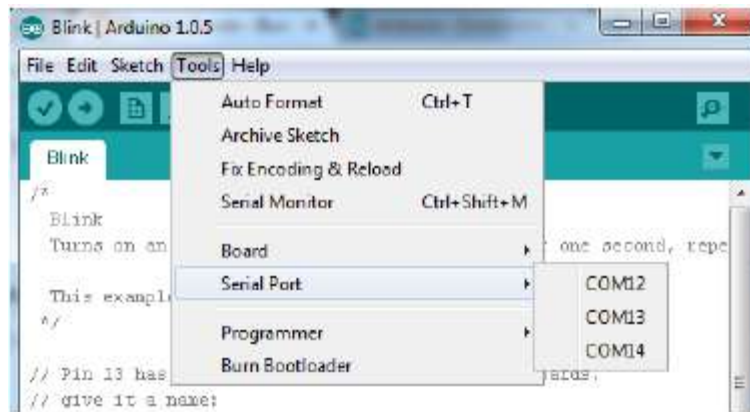
Gambar 2.6. Contoh Blink LED

- Pilih Board yang digunakan dengan memilih entri dalam menu Tools > Board yang sesuai dengan Arduino. Misalnya dipilih Arduino Uno (Gambar 2.7).



Gambar 2.7. Memilih Board

- Pilih port serial (Gambar 2.8)
Pilih perangkat serial Arduino dari menu Tools | Serial Port. Biasanya Port akan otomatis terdeteksi.



Gambar 2.8. Memilih Port

- Upload Program (Gambar 2.9)
 Dengan meng-klik tombol "Upload". Tunggu beberapa detik – akan terlihat akan berkedip LED RX dan TX pada Arduino Board.



Gambar 2.9. Upload Program

2.3. Ringkasan

1. Arduino merupakan papan-tunggal mikrokontroler serba guna yang bisa diprogram dan bersifat *open-source*.
2. Arduino Uno adalah salah satu kit mikrokontroler yang berbasis pada **ATmega328**. Modul ini sudah dilengkapi dengan berbagai hal yang dibutuhkan untuk mendukung mikrokontroler untuk bekerja.
3. Arduino IDE menggunakan bahasa pemrograman **C++** dengan versi yang telah disederhanakan, sehingga lebih mudah dalam belajar pemrograman.
4. Software Arduino IDE (*Integrated Development Environment*) adalah sebuah perangkat lunak yang memudahkan dalam mengembangkan aplikasi mikrokontroler mulai dari menuliskan source program, kompilasi, upload hasil kompilasi, dan uji coba secara terminal serial.
5. Hubungan komunikasi data antara IDE arduino dengan board Arduino digunakan komunikasi secara serial dengan protokol RS232. Jika board arduino sudah dilengkapi dengan komunikasi serial RS232 (biasanya USB), maka dapat langsung ditancapkan ke USB komputer.

3. PEMROGRAMAN ARDUINO

3.1. Konsep Dasar Pemrograman Arduino

Software/program yang ditulis menggunakan Arduino disebut sketsa (*sketch*). Sketsa ini ditulis dalam editor teks. Sketsa disimpan dengan ekstensi file `.ino`, yang memiliki fitur untuk meng-*cut*, meng-*copy*, mem-*paste*, mencari/ mengganti teks, dll. Area pesan (*console*) memberikan umpan balik, menyimpan dan mengekspor juga menampilkan kesalahan. Konsol menampilkan teks output dalam lingkungan Arduino termasuk detail pesan error dan informasi lainnya. Bagian sudut bawah sebelah kanan jendela menampilkan board dan port serial yang sedang dipakai. Tombol-tombol toolbar (Gambar 3.1.) memungkinkan untuk memverifikasi dan meng-*upload* program, membuat, membuka, dan menyimpan sketsa, juga membuka monitor serial.



Gambar 3.1. Toolbar dan Menu Arduino

Code Program Arduino biasa disebut *sketch* dan dibuat menggunakan bahasa pemrograman C. Program atau sketch yang sudah selesai ditulis di Arduino IDE dapat langsung di-*compile* dan di-*upload* ke Arduino Board.


```
sketch_feb20a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Gambar 3.2. Sketch dalam Arduino IDE

Secara sederhana, *sketch* dalam Arduino dikelompokkan menjadi 3 blok (Gambar 3.2.), yakni : Header, Setup dan Loop. Untuk program yang lebih kompleks akan ada blok lain berupa fungsi-fungsi pendukung.

Header

Pada bagian ini biasanya ditulis definisi-definisi penting yang akan digunakan selanjutnya dalam program, misalnya penggunaan library dan pendefinisian variable. Code dalam blok ini dijalankan hanya sekali pada waktu compile. Di bawah ini contoh code untuk mendeklarasikan variable led (integer) dan sekaligus di isi dengan angka 13

```
int led = 13;
```

Setup

Di sinilah awal program Arduino berjalan, yaitu di saat awal, atau ketika power on Arduino board. Biasanya di blok ini diisi penentuan apakah suatu pin digunakan sebagai input atau output, menggunakan perintah pinMode. Inisialisasi variable juga bisa dilakukan di blok ini

```
// the setup routine runs once when you press reset:

void setup() { // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
```

OUTPUT adalah suatu makro yang sudah didefinisikan Arduino yang berarti = 1. Jadi perintah di atas sama dengan `pinMode(led, 1);`

Suatu pin bisa difungsikan sebagai OUTPUT atau INPUT. Jika difungsikan sebagai output, dia siap mengirimkan arus listrik (maksimum 100 mA) kepada beban yang disambungkannya. Jika difungsikan sebagai INPUT, pin tersebut memiliki impedance yang tinggi dan siap menerima arus yang dikirimkan kepadanya.

Loop

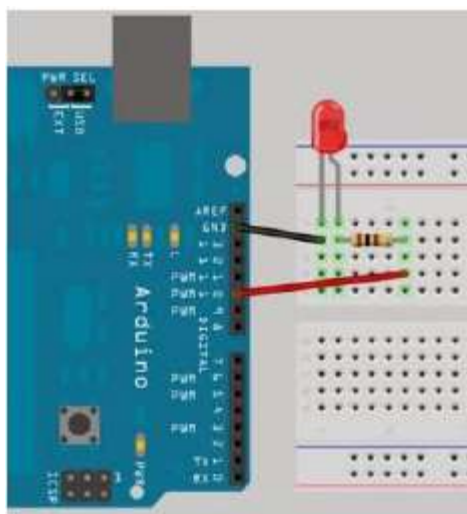
Blok ini akan dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok. Program akan berhenti apabila tombol power Arduino di matikan. Di sinilah fungsi utama program Arduino kita berada.

```
void loop() {  
    digitalWrite(led, HIGH); // nyalakan LED  
    delay(1000); // tunggu 1000 milidetik  
    digitalWrite(led, LOW); // matikan LED  
    delay(1000); // tunggu 1000 milidetik  
}
```

Perintah `digitalWrite(pinNumber, nilai)` akan memerintahkan arduino untuk menyalakan atau mematikan tegangan di `pinNumber` tergantung nilainya. Jadi perintah di atas `digitalWrite(led, HIGH)` akan membuat pin nomor 13 (karena di header dideklarasikan `led = 13`) memiliki tegangan = 5V (HIGH). Hanya ada dua kemungkinan nilai `digitalWrite` yaitu HIGH atau LOW yang sebetulnya adalah nilai integer 1 atau 0

3.2. Program Sederhana Arduino

Program sederhana adalah *Led Blink*, program ini akan mengakses pin 10 dan memerintahkan Arduino untuk mengulang *blink led*, Gambar 3.3.



Gambar 3.3. Wiring LED blinking

Sketch

```
// Project 1 - LED Flasher
int ledPin = 10;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

Pembahasan Sketch

```
// Project 1 - LED Flasher
```

Ini adalah komentar baris yang berguna untuk dokumentasi program, kompiler akan mengabaikan bagian ini. Baris komentar berguna bagi programmer agar bisa mengerti maksud program.

```
int ledPin = 10;
```

Inisialisasi variable, dalam hal ini inisialisasi variable bernama `ledPin` dengan type data integer dan nilai 10.

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

Setiap sketch Arduino wajib memiliki fungsi `setup()` dan `loop()`. Fungsi `setup()` dipanggil hanya sekali saat pertama kali program berjalan. Fungsi `setup()` biasanya tempat untuk men-setup hal-hal umum agar program siap dijalankan, seperti setup pin modes, setting serial baud rates, dan lainnya. Pada sketch Led Blink, fungsi setup hanya memiliki 1 baris perintah yaitu

```
pinMode(ledPin, OUTPUT);
```

`pinMode` fungsi yang berguna untuk memberitahu arduino bahwa pin pada board akan digunakan sebagai input atau output. Dalam baris program di atas, memberitahu arduino untuk menset pin 10 (nilai `ledPin` adalah 10) sebagai Output.

```
void loop() {  
    digitalWrite(ledPin, HIGH);  
    delay(1000);  
    digitalWrite(ledPin, LOW);  
    delay(1000);  
}
```

Fungsi `loop()` function adalah program utama yang dipanggil secara continue selama Arduino menyala. Setiap perintah dalam fungsi `loop()` akan dipanggil satu persatu sampai perintah terakhir dalam blok loop dicapai, lalu Arduino akan kembali ke awal perintah di blok fungsi `loop()`, sampai Arduino dimatikan atau tombol reset ditekan.

3.3. Ringkasan

1. Code Program Arduino biasa disebut *sketch* dan dibuat menggunakan bahasa pemrograman C. Program atau sketch yang sudah selesai ditulis di Arduino IDE dapat langsung di-*compile* dan di-*upload* ke Arduino Board.
2. Secara sederhana, *sketch* dalam Arduino dikelompokkan menjadi 3 blok yakni : Header, Setup dan Loop.
3. Setiap sketch Arduino wajib memiliki fungsi `setup()` dan `loop()`. Fungsi `setup()` dipanggil hanya sekali saat pertama kali program berjalan. Fungsi `loop()` function adalah program utama yang dipanggil secara continue selama Arduino menyala.

DAFTAR BACAAN

1. John Crisp, 2004, Introduction Microprocessors and Microcontrollers (2nd Edition), an imprint of Elsevier, ISBN: 0-7506-5989-0
2. Jack Purdum, 2012, Beginning C for Arduino, Published by Springer Science, ISBN: 978-1-4302-4777-7
3. Michael Margolis, 2011, Arduino Cookbook, Published by O'Reilly Media, Inc., ISBN: 978-0-596-80247-9
4. John Boxall, 2013, Arduino Workshop a Hands-on Introduction with 65 Projects, No Starch Press, Inc, ISBN: 978-1-59327-448-1.
5. Dr. Muhammad Yusro, MT, Modul Teori dan Praktikum “, Program Studi D3 Teknik Elektronika Fakultas Teknik Universitas Negeri Jakarta 2017.
6. Sumber internet lainnya :
 - <https://www.arduino.cc/>
 - <http://www.arduino.web.id/>
 - <http://elektrokita.com/mengenal-arduino-uno/>
 - <http://panduanbelajarmicrocontroller.blogspot.co.id/2013/04/pengantar-belajar-arduino.html>
 - <http://aozon.blogspot.co.id/2014/03/perkenalanmengenal-arduino-dari-pemula.html>
 - <http://bsiswoyo.lecture.ub.ac.id/2012/06/belajar-arduino-pengantar/>
 - <http://arduino-elektro-robotik.blogspot.co.id/2014/01/05-project-led-blink.html>
 - <http://www.rustamaji.net/id/component/arduino-uno>

BAB II PRAKTIKUM

Tgl Percobaan	Judul Percobaan 1	Paraf
	LED Blink	

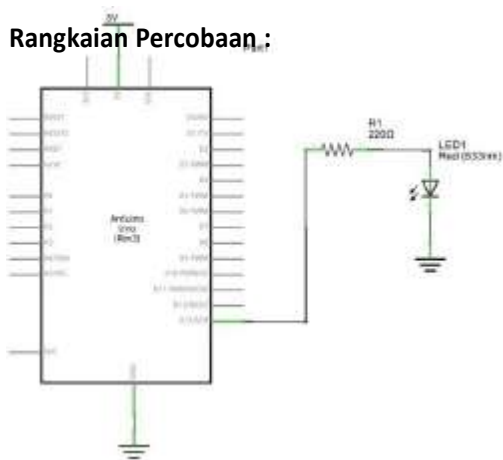
A. Tujuan Percobaan :

1. Menggunakan Arduino sebagai output.
2. Membuat program sederhana untuk membuat LED pada pin 13 berkedip.

B. Alat dan Bahan :

Arduino Board dengan LED built-in.

C. Rangkaian Percobaan :



D. Program Percobaan :

```
int ledPin = 13; // LED connected to digital pin 13
void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000); // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(1000); // waits for a second
}
```

Tgl Percobaan	Judul Percobaan 2	Paraf
	Push Button	

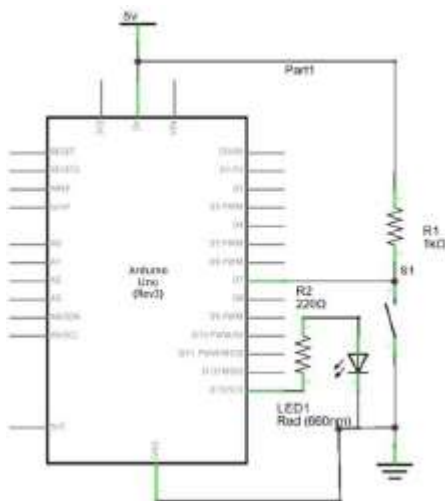
A. Tujuan Percobaan :

1. Menggunakan Arduino sebagai input
2. Membuat program sederhana dengan push button sebagai input untuk menyalakan dan mematikan nyala lampu LED.

B. Alat dan Bahan :

1. Arduino Board dengan LED built- in.
2. Push Button
3. Resistor 10k Ohm

C. Rangkaian Percobaan :



D. Program Percobaan :

```

int ledPin = 13; // choose the pin for the LED
int inPin = 2; // choose the input pin (for a pushbutton)
int val = 0; // variable for reading the pin status
void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inPin, INPUT); // declare pushbutton as input
}
void loop(){
  val = digitalRead(inPin); // read input value
  if (val == HIGH) { // check if the input is HIGH (button released)
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}

```

Tgl Percobaan	Judul Percobaan 3	Paraf
	Loop	

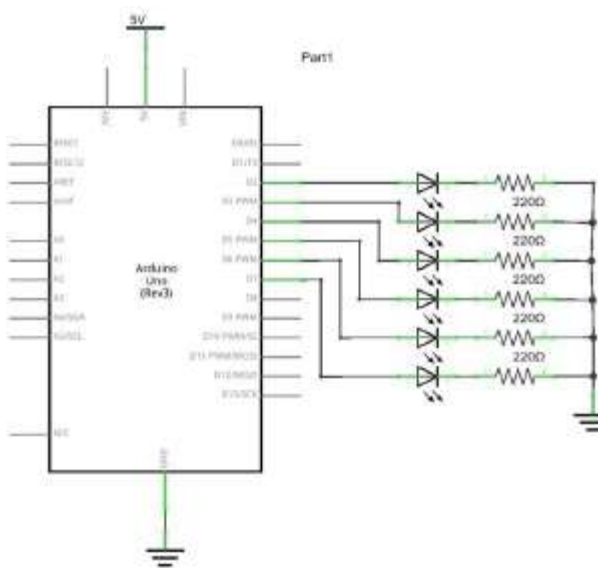
A. Tujuan Percobaan :

1. Membuat program dengan teknik loop dalam arduino untuk aplikasi running LED
2. Merangkai rangkaian aplikasi running LED dengan arduino.

B. Alat dan Bahan :

4. Arduino Board
5. Bread Board
6. Kawat Jumper
7. Resistor 330 Ohm6 buah
8. LED 6 buah

C. Rangkaian Percobaan :



D. Program Percobaan :

```
int timer = 500; // The higher the number, the slower the timing.
int pins[] = { 2, 3, 4, 5, 6, 7 }; // an array of pin numbers
int num_pins = 6; // the number of pins (i.e. the length of the array)

void setup()
{
  int i;
  for (i = 0; i < num_pins; i++){ // the array elements are numbered from 0 to num pins - 1
    pinMode(pins[i], OUTPUT); // set each pin as an output
  }
}

void loop()
{
  int i;
```



```
for (i = 0; i < num_pins; i++) { // loop through each pin...
digitalWrite(pins[i], HIGH); // turning it on,
delay(timer); // pausing,
digitalWrite(pins[i], LOW); // and turning it off.
}
for (i = num_pins - 1; i >= 0; i--) {
digitalWrite(pins[i], HIGH);
delay(timer);
digitalWrite(pins[i], LOW);
}
}
```

Tgl Percobaan	Judul Percobaan 4	Paraf
	Analog Input	

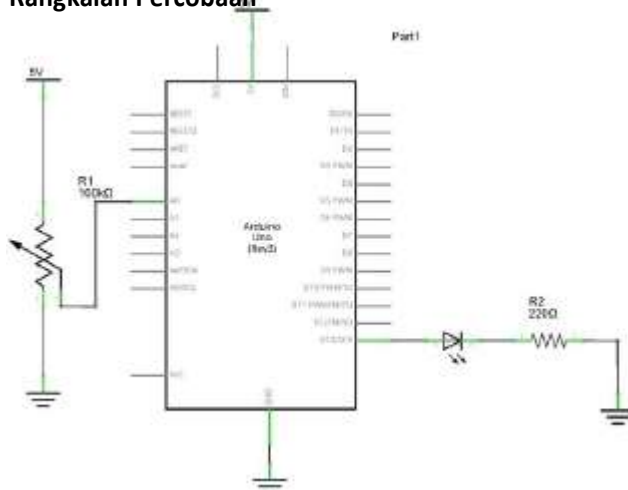
A. Tujuan Percobaan :

1. Menggunakan Analog input pada arduino
2. Membuat program sederhana dengan menggunakan analog input sebagai delay untuk lama nyala LED.

B. Alat dan Bahan :

1. Arduino Uno
2. Potensio Meter
3. Bread Board
4. Kawat Jumper

C. Rangkaian Percobaan



E. Program Percobaan :

```

int potPin = 2; // select the input pin for the potentiometer
int ledPin = 13; // select the pin for the LED
int val = 0; // variable to store the value coming from the sensor
void setup() {
  pinMode(ledPin, OUTPUT); // declare the ledPin as an OUTPUT
}
void loop() {
  val = analogRead(potPin); // read the value from the sensor
  digitalWrite(ledPin, HIGH); // turn the ledPin on
  delay(val); // stop the program for some time
  digitalWrite(ledPin, LOW); // turn the ledPin off
  delay(val); // stop the program for some time
}

```

Tgl Percobaan	Judul Percobaan 5	Paraf
	LCD	

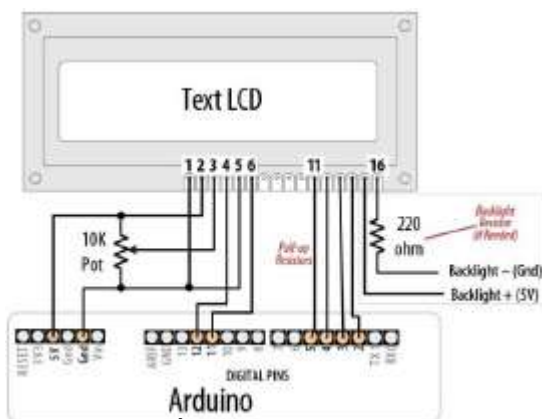
A. Tujuan Percobaan :

1. Menggunakan LCD pada Arduino
2. Membuat program sederhana untuk menampilkan text pada LCD 16 x2

B. Alat dan Bahan :

1. Arduino Uno
2. LCD 16 x2
3. Kawat jumper
4. Bread board

C. Rangkaian Percobaan :



D. Program Percobaan :

```
#include <LiquidCrystal.h>

const int PIN_RS      = 12;
const int PIN_E       = 11;
const int PIN_DB_4    = 5;
const int PIN_DB_5    = 4;
const int PIN_DB_6    = 3;
const int PIN_DB_7    = 2;

// Buat objek LCD
LiquidCrystal lcd(PIN_RS, PIN_E, PIN_DB_4,
                 PIN_DB_5, PIN_DB_6, PIN_DB_7);

void setup()
{
  // Tentukan ukuran LCD
  lcd.begin(16, 2);
}

void loop()
{
```

```
lcd.clear();  
lcd.print("1 + 1 ada brapa?");  
lcd.setCursor(0, 1);  
lcd.print("Dari dulu juga 2");  
  
delay(5000);  
  
lcd.clear();  
lcd.print("Mahu tidak mahu");  
lcd.setCursor(0, 1);  
lcd.print("Trus maju selalu");  
  
delay(5000);  
}
```

Tgl Percobaan	Judul Percobaan 6	Paraf
	Unipolar Motor Stepper	

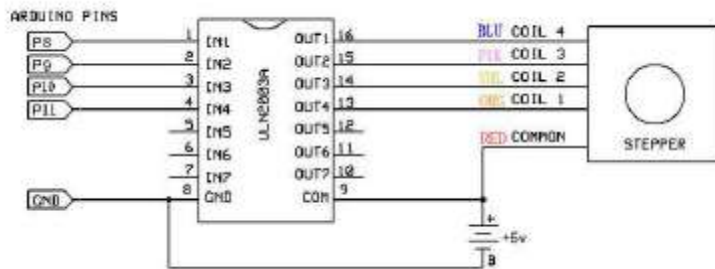
A. Tujuan Percobaan :

1. Menggunakan motor stepper pada Arduino.
2. Membuat program sederhana untuk menggerakkan motor stepper.

B. Alat dan Bahan :

3. Arduino Uno
4. Motor Stepper
5. ULN 2003
6. Bread board
7. Kawat Jumper

C. Rangkaian Percobaan :



D. Program Percobaan :

```
#include <Stepper.h>
const int STEP_SATU_PUTARAN = 2048;
// Buat objek motor stepper
Stepper motorStepper(STEP_SATU_PUTARAN, 8, 9, 10, 11);

void setup()
{
  // Atur kecepatan
  motorStepper.setSpeed(4);
}

void loop()
{
  // Putar searah jarum jam
  motorStepper.step(STEP_SATU_PUTARAN);
  delay(1000);

  // Putar berlawanan arah jarum jam
  motorStepper.step(-STEP_SATU_PUTARAN);
  delay(1000);
}
```

Tgl Percobaan	Judul Percobaan 7	Paraf
	Keypad	

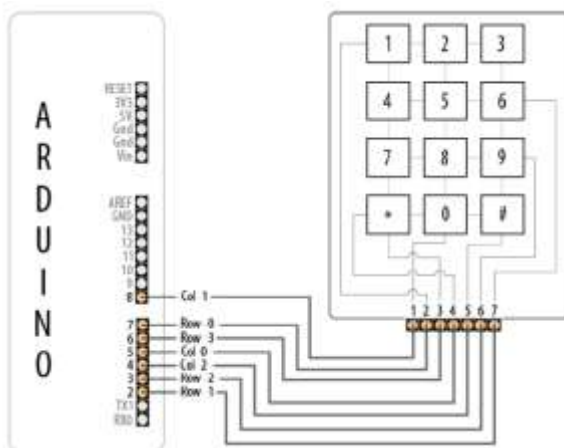
A. Tujuan Percobaan :

1. Menggunakan keypad sebagai input pada arduino
2. Membuat program sederhana untuk menampilkan input keypad di serial monitor.

B. Alat dan Bahan :

1. Arduino Uno
2. Keypad
3. Bread board
4. Kawat Jumper

C. Rangkaian Percobaan :



Program Percobaan :

```

const int numRows = 4; // number of rows in the keypad
const int numCols = 3; // number of columns
const int debounceTime = 20; // number of milliseconds for switch to be stable

const char keymap[numRows][numCols] = {
  { '1', '2', '3' },
  { '4', '5', '6' },
  { '7', '8', '9' },
  { '*', '0', '#' }
};

// this array determines the pins used for rows and columns
const int rowPins[numRows] = { 7, 2, 3, 6 }; // Rows 0 through 3
const int colPins[numCols] = { 5, 8, 4 }; // Columns 0 through 2

void setup()

```

```

{
Serial.begin(9600);
for (int row = 0; row < numRows; row++)
{
pinMode(rowPins[row],INPUT); // Set row pins as input
digitalWrite(rowPins[row],HIGH); // turn on Pull-ups
}
for (int column = 0; column < numCols; column++)
{
pinMode(colPins[column],OUTPUT); // Set column pins as outputs for writing
digitalWrite(colPins[column],HIGH); // Make all columns inactive
}
}

void loop()
{
char key = getKey();
if( key != 0) { // if the character is not 0 then it's a valid key press
Serial.print("Got key ");
Serial.println(key);
}
}

// returns with the key pressed, or 0 if no key is pressed

char getKey()
{
char key = 0; // 0 indicates no key pressed
for(int column = 0; column < numCols; column++)
{
digitalWrite(colPins[column],LOW); // Activate the current column.
for(int row = 0; row < numRows; row++) // Scan all rows for a key press.
{
if(digitalRead(rowPins[row]) == LOW) // Is a key pressed?
{
delay(debounceTime); // debounce
while(digitalRead(rowPins[row]) == LOW)
; // wait for key to be released
key = keypad[row][column]; // Remember which key was pressed.
}
}
digitalWrite(colPins[column],HIGH); // De-activate the current column.
}
return key; // returns the key pressed or 0 if none
}

```

Tgl Percobaan	Judul Percobaan 8	Paraf
	Ultrasonik	

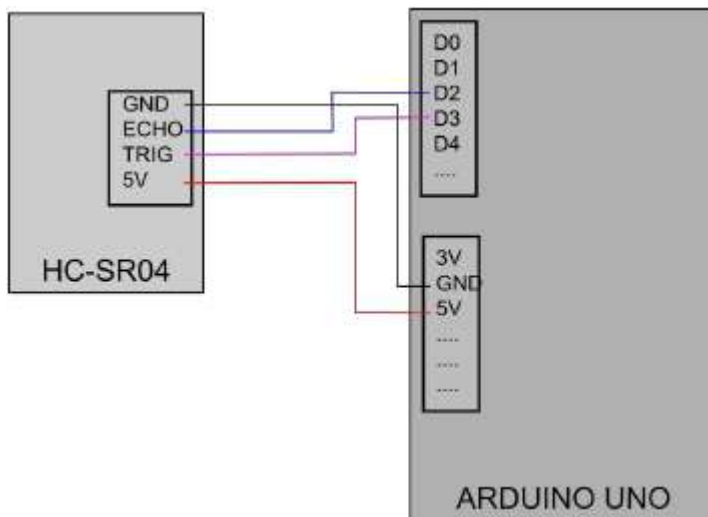
A. Tujuan Percobaan :

1. Menggunakan Ultrasonik HC SR04 pada Arduino.
2. Membuat program sederhana untuk membaca jarak dengan modul ultrasonik HC SR04.

B. Alat dan Bahan :

1. Arduino Uno.
2. Ultrasonik.
3. Kawat jumper.

C. Rangkaian Percobaan :



Program Percobaan :

```
#define trigPin 3
#define echoPin 2
#define led 13

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

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW); // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin, HIGH);
  // delayMicroseconds(1000); - Removed this line
  delayMicroseconds(10); // Added this line
```



```
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
if (distance < 4) { // This is where the LED On/Off happens
digitalWrite(led,HIGH); // When the Red condition is met, the Green LED should turn
off
}
else {
digitalWrite(led,LOW);
}
if (distance >= 200 || distance <= 0){
Serial.println("Out of range");
}
else {
Serial.print(distance);
Serial.println(" cm");
}
delay(500);
}
```