

MODUL PRAKTIKUM
SISTEM OTOMASI & *ROBOTIC*



LABORATORIUM SISTEM OTOMASI
PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNIK
UNIVERSITAS KRISNADWIPAYANA
JAKARTA

2022

KATA PENGANTAR

Puji syukur dengan karuniaNya sehingga penulis dapat menyelesaikan modul ajar yang berjudul Praktikum Sistem Otomasi dan *Robotic*. Modul ini disusun dengan tujuan untuk meningkatkan pemahaman praktikan tentang Sistem Otomasi dan *Robotic*.

Struktur modul ini terdiri dari tujuan praktikum, uraian materi yang ada didalam praktikum, dan pengaplikasian materi yang ada didalam praktikum tersebut. Tujuan pembelajaran digunakan untuk mengetahui arah atau tujuan mempelajari materi tertentu. Uraian materi digunakan untuk pemberian informasi/ pengetahuan kepada praktikan. Uraian materi tersebut meliputi pemrograman plc dan mikrokontroler.

Apa yang telah dituangkan dalam tulisan ini tidak luput dari kesalahan baik disengaja ataupun tidak disengaja, sehingga penulis menerima kritik dan saran yang bersifat membangun yang diberikan untuk evaluasi penulisan modul praktikum ini. Besar harapan penulis agar hasil dari penulisan modul praktikum Sistem Otomasi dan *Robotic* ini dapat bermanfaat.

Jakarta, 20 November 2022

Tim Penyusun Modul Praktikum

TATA TERTIB PRAKTIKUM

1. Peserta praktikum harus datang tepat waktu yang telah ditentukan 15menit sebelum praktikum dimulai.
2. Tidak diperkenankan mengikuti kegiatan praktikum selambat-lambatnya 15 menit setelah praktikum berjalan sesuai jadwal yang ditentukan.
3. Peserta praktikum harus menggunakan pakaian rapi dan sopan.
4. Peserta praktikum diwajibkan menggunakan masker dan saling menjaga jarak mengikuti aturan Pemerintah.
5. Sebelum memasuki ruangan Praktikum peserta diwajibkan menunjukkan bukti status di SIPRAM.
6. Sebelum mengikuti praktikum, seluruh peserta praktikum harus mengumpulkan tugas pendahuluan sesuai tanggal yang telah ditetapkan dan mengisi daftar hadir.
7. Selama praktikum, praktikan harus menjaga kebersihan tempat.
8. Praktikan tidak diperkenankan merokok, makan, membuang sampah sembarangan, membuat keributan atau mengganggu ketertiban selama kegiatan praktikum berlangsung.
9. Seluruh praktikan dilarang meninggalkan tempat praktikum tanpa izin asisten lab.
10. Praktikan wajib turut serta menjaga peralatan lab. Kerusakan atau kehilangan peralatan yang disebabkan karena kelalaian praktikan merupakan tanggung jawab seluruh praktikan.
11. Menghadirkan pengarahannya modul praktikum untuk memudahkan laporan.
12. Asistensi dilaksanakan sesuai dengan waktu yang telah disepakati.
13. Menghadiri praktikum dan bila tidak mengikuti kegiatan praktikum atau tidak hadir maka dianggap yang bersangkutan mengundurkan diri.
14. Dispensasi untuk kegiatan laboratorium hanya diberikan sesuai dengan peraturan jurusan Teknik Industri, yaitu:
 - a. Mahasiswa yang di opname di rumah sakit yang disertakan bukti.
 - b. Mahasiswa yang ditugaskan oleh universitas krisnadwipayana yang disertakan dengan bukti.
 - c. Orang tua atau saudara meninggal yang disertakan bukti.
 - d. Surat bukti yang dibutuhkan paling lambat diserahkan 3 hari dari praktik.

IDENTITAS PRAKTIKUM

- 1) Nama Mata Praktikum : Sistem Otomasi
- 2) Pelaksanaan : Semester Ganjil
- 3) Kepala Laboratorium : Aries Abbas, S.T, M.M, M.T
- 4) Asisten Laboratorium : 1. Ainul Rizqi
2. Brilliano Aprilianto
3. Dimas Yugimurcito
4. Dwi Nurfadillah Putri

1. Tujuan Umum Praktikum

Pada pertemuan ini akan dibahas mengenai Konsep Sistem Otomasi . Setelah menyelesaikan praktikum, mahasiswa mampu memahami dan menjabarkan tentang konsep Sistem Otomasi serta dapat mengaplikasikannya.

2. Tempat pelaksanaan praktikum

Tempat berlangsungnya kegiatan praktikum ini adalah di Laboratorium Sistem Otomasi Fakultas Teknik Universitas Krisnadwipayana.

3. Alokasi waktu praktikum

Waktu praktikum yang dibutuhkan untuk satu kali pertemuan tatap muka ialah 90 menit.

4. Evaluasi dan Penilaian Hasil Belajar Mahasiswa Nilai praktikum diambil dari:

Tugas Pendahuluan	: 10%
Pelaksanaan Praktikum	: 20%
Asistensi	: 15%
Responsi	: 15%
Laporan Akhir	: 40%

FORMAT LAPORAN PRAKTIKUM

A. Format Laporan

1. Laporan diketik menggunakan kertas HVS berukuran A4 80 gram dan dijilid berwarna “Hijau”
2. Menggunakan batas margin:
 - Kiri : 3,5 cm
 - Kanan : 3 cm
 - Atas : 3 cm
 - Bawah : 3 cm
3. Menggunakan *line spacing* 2,0.
4. Menggunakan italic pada Bahasa asing
5. Format pengetikan menggunakan font “*Times New Roman*” dengan format ukuran sebagai berikut:
 - Judul Bab : 14, Bold
 - Sub Bab : 12, Bold
 - Isi : 12
 - Nomor Halaman : 10
6. Format penempatan nomor halaman:
 - Judul Bab : Tengah bagian Bawah
 - Isi : Kanan bagian Atas
7. Format tabel dan gambar:
 - Tabel : Di atas, Perbab
 - Gambar : Di bawah, Perbab
8. Format daftar pustaka:
 - Jurnal & Buku : Nama - Tahun - Judul - Kota – Penerbit.
 - Website* : Salin *link*.

B. Susunan Laporan

Cover

Lembar Pengesahan

Kata Pengantar

Daftar Isi

Daftar Gambar

Daftar Tabel

BAB I Pendahuluan

1.1 Latar Belakang Masalah

1.2 Rumusan Masalah

1.3 Tujuan dan Manfaat

1.4 Batasan Masalah

1.5 Flowchart Pemecahan Masalah

1.6 Sistematika Penulisan

BAB II Landasan Teori (minimal 10 halaman)

BAB III Pengumpulan Dan Pengolahan Data

BAB IV Analisis

BAB V Kesimpulan Dan Saran

Daftar Pustaka (minimal 2 Jurnal / Buku)

LEMBAR PENGESAHAN

Laporan praktikum ini telah dibuat oleh :

Nama :

NIM :

Program Studi :

Fakultas/Universitas :

Waktu pelaksanaan :

Judul :

Telah diperiksa dan disetujui untuk diberikan penilaian yang sebagaimana mestinya.

Menyetujui,

Kepala Laboratorium

Asisten Laboratorium

(Nama & Gelar)

NIDN :

(Nama)

NIM :

Mengetahui,

Kepala Program Studi

Ir. Florida Butarbutar, MT

NIDN : 0310056507

MODUL 1

PROGRAMMABLE LOGIC CONTROLLER (PLC)



**LABORATORIUM SISTEM OTOMASI
PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNIK
UNIVERSITAS KRISNADWIPAYANA
JAKARTA**

2022

MODUL 1

PROGRAMMABLE LOGIC CONTROLLER (PLC)

A. LATAR BELAKANG

Otomasi industri adalah proses di mana sistem kontrol menggantikan peran manusia dalam industri. Artinya, banyak tahap dalam proses industri tidak lagi dijalankan manusia. Melainkan digantikan oleh mesin sehingga keseluruhan proses menjadi otomatis.

Demi memberlakukan otomasi, industri memerlukan sebuah sistem kontrol. Sistem tersebut harus dapat memastikan proses manufaktur yang efektif dan efisien. Karena itulah sekarang ada banyak jasa kontraktor otomasi yang menawarkan sistem kontrol dengan kualitas mumpuni.

Sistem yang digunakan dalam otomasi industri meliputi penggunaan perangkat kontrol untuk mengendalikan proses operasi. Perangkat tersebut memungkinkan proses operasi untuk dikontrol secara otomatis tanpa memerlukan banyak peran manusia.

Beberapa contoh perangkat kontrol yaitu komputer industri atau *industrial PC*, *programmable logic controller (PLC)*, dan *distributed control system (DCS)*. Walaupun sistemnya berbeda-beda, namun semua perangkat tersebut memiliki tujuan yang sama yaitu menghasilkan output yang diinginkan.

Sistem otomasi memiliki banyak keunggulan. Salah satunya adalah biaya produksi yang menurun. Dalam jangka panjang, menggantikan tenaga manusia dengan mesin dapat memotong biaya produksi cukup banyak. Produktivitas dan profit pun meningkat karena otomasi. Sebab, operasi manufaktur dapat dilakukan dengan lebih cepat dan akurat.

B. Tujuan Praktikum

Tujuan diadakan nya praktikum ini adalah agar peserta mampu :

1. Mempelajari cara memprogram PLC dengan diagram *ladder*
2. Memprogram PLC dengan *ladder* untuk mengaktifkan *button* pada PLC, menjalankan *conveyer* dan sensor.

C. Dasar teori

1. Defisini PLC (*Programmable Logic Controller*)



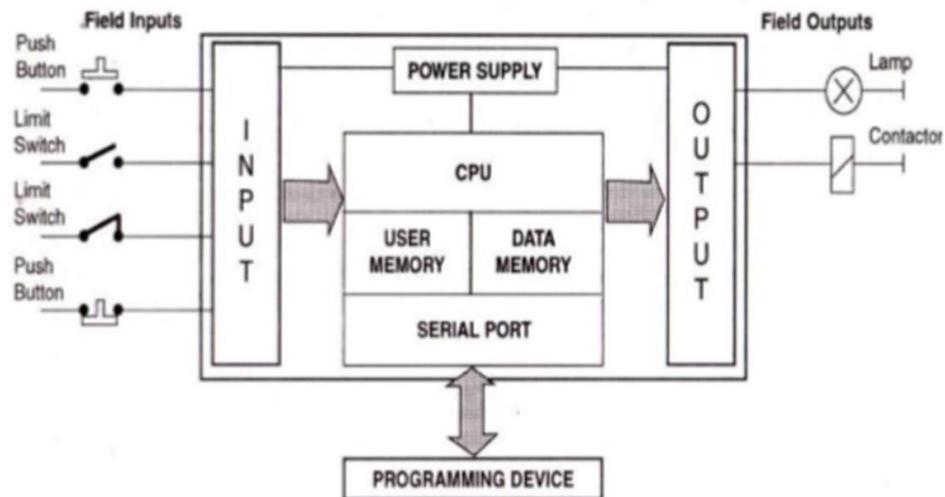
Gambar 1. 1 Jenis-jenis PLC

PLC (*Programmable Logic Controller*) dapat diartikan sebagai kontrol logika terprogram. PLC memiliki "otak" berupa mikroprosesor, digunakan pada otomasi proses industri seperti pengawasan dan pengontrolan mesin- mesin produksi. PLC memiliki perangkat masukan dan keluaran yang digunakan untuk berhubungan dengan perangkat luar seperti tombol operasi, sensor, relai, *contactor*, dll. Bahasa pemrograman yang digunakan untuk mengoperasikan PLC berbeda dengan bahasa pemrograman biasa. Bahasa yang digunakan adalah *Ladder*, yang hanya berisi input-proses-output. Disebut *Ladder*, karena bentuk tampilan bahasa

pemrogramannya memang seperti tampilan tangga. Disamping menggunakan pemrograman *ladder*, PLC juga dapat diprogram dengan pemrograman SFC dan pemrograman FBD.

2. Komponen Utama PLC

Komponen utama PLC adalah: (1) *Input*, (2) CPU (*Central Processing Unit*), dan (3) *Output*. *Input* pada PLC bisa berupa alat untuk mengoperasikan sistem (saklar, tombol) dan sensor. *Output* pada PLC adalah sistem yang dikontrol, bisa berupa aktuator (motor, kontaktor), lampu dan sebagainya. Penempatan terminal *Input* dan terminal *Output* pada PLC merk apapun selalu terpisah jauh (berseberangan). Hal ini dimaksudkan untuk mempermudah merangkai dan memperkecil terjadinya kesalahan merangkai. Contoh jika terminal *input* diletakkan di bagian atas PLC, maka terminal *output* diletakkan di bagian bawah PLC tersebut. Ada juga merk PLC yang *input output*nya terpisah, yakni ditempatkan pada modul tambahan khusus *input* dan modul tambahan yang lain untuk *output*.



Gambar 1. 2 Komponen utama PLC

3. Aplikasi yang Digunakan

➤ CX Programmer

4. Contoh Program Sederhana: Start-Stop Lampu

KASUS: Sebuah lampu dinyalakan melalui tombol *START* dan dimatikan melalui tombol *STOP*.

ALGORITMA KONTROL

- Tekan tombol *START* , maka lampu akan menyala
- Tekan tombol *STOP*, maka lampu akan mati.

NO	PERALATAN INPUT	ALAMAT	PERALATAN OUTPUT	ALAMAT
1	Tombol START	0.00	LAMPU	100.00
2	Tombol STOP	0.01		



Gambar 1. 3 contoh program sederhana

5. Mengenal *Software* CX-Programmer 9.1

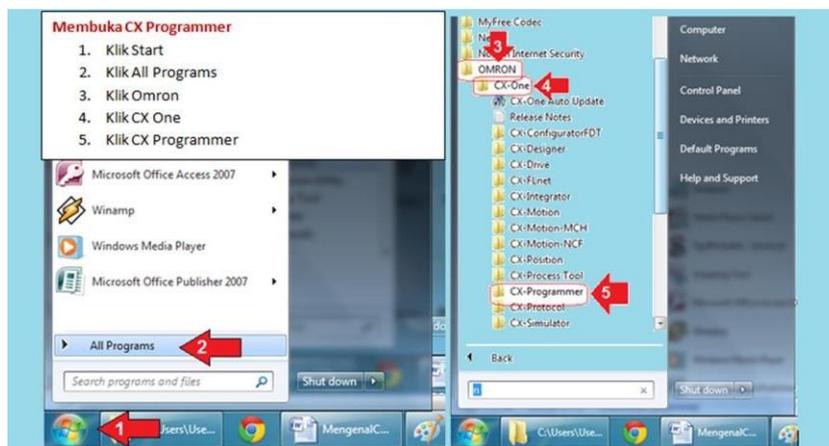
CX-Programmer adalah software aplikasi yang dikembangkan oleh Omron untuk memprogram semua jenis PLC produk Omron. Modul ini hanya membahas pemrograman menggunakan CX-Programmer versi 9.1

- Memulai CX-Programmer

CX-Programmer adalah *software* aplikasi berbasis Windows. Oleh karena itu mengaktifkannya mirip dengan *software* berbasis Windows lainnya. Beberapa pilihan cara mengaktifkannya adalah sebagai berikut:

Alternatif pertama:

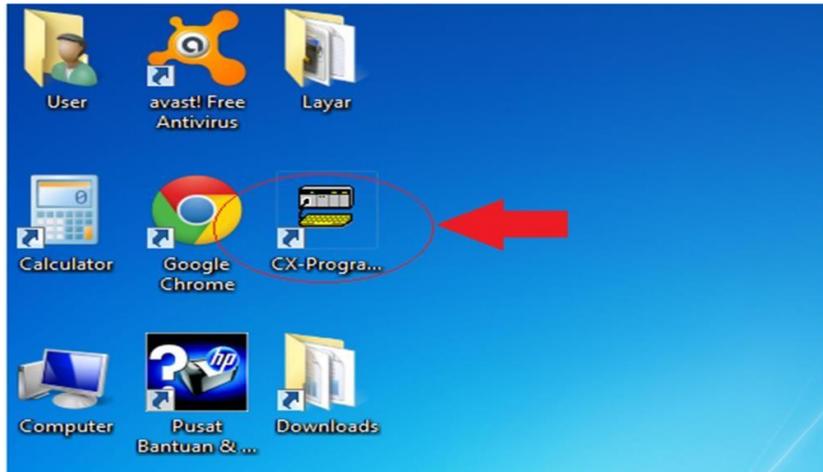
1. Klik *Start*
2. Klik *All Programs*
3. Klik Omron
4. Klik CX One
5. Klik CX Programmer, maka akan muncul tampilan awal seperti gambar dibawah



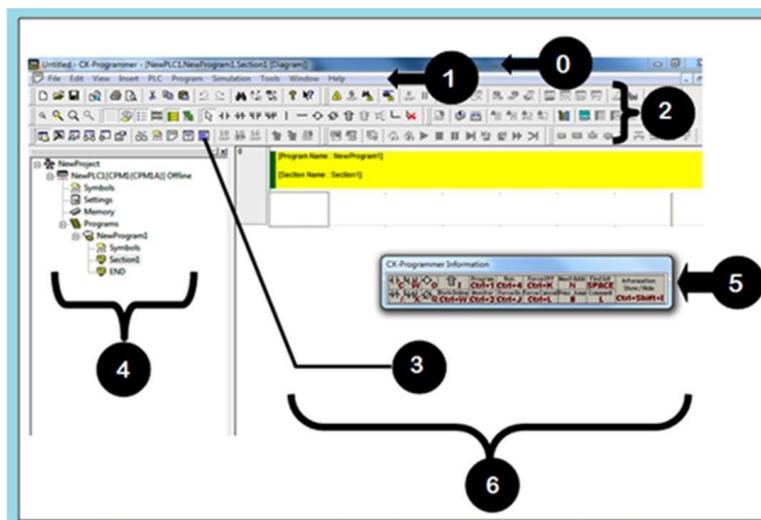
Gambar 1. 4 Tampilan awal

Alternatif kedua:

Jika pada *desktop* ada icon  maka tinggal *double* klik pada gambar *icon* tersebut.



Gambar 1. 5 Tampilan desktop



- Keterangan**
- 0. Titlebar
 - 1. Menubar
 - 2. Tool
 - 3. ICON
 - 4. Proyek yang sedang dikerjakan
 - 5. Daftar Tombol Shortcut
 - 6. Bidang kerja

Gambar 1. 6 Bagian layar CX-Programmer V 9.1

Tombol *Shortcut*

Tombol *Shortcut* adalah tombol yang digunakan untuk membuat komponen *Ladder*.

C : membuat tombol *Normaly Open*

/ : membuat tombol *Normaly Close*

W : membuat tombol *Normaly Open OR*

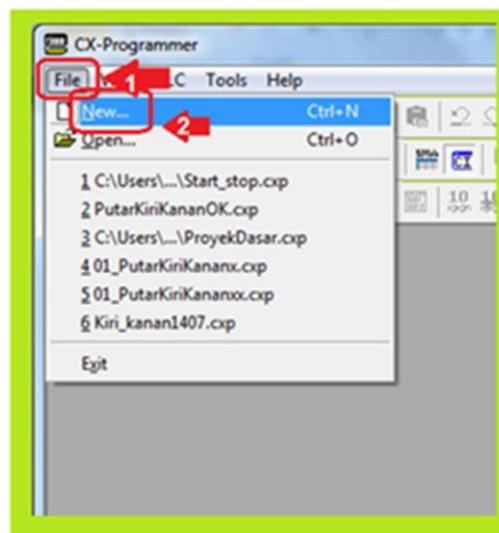
X : membuat tombol *Normaly Close OR*

O : membuat *Normal Open Coil*

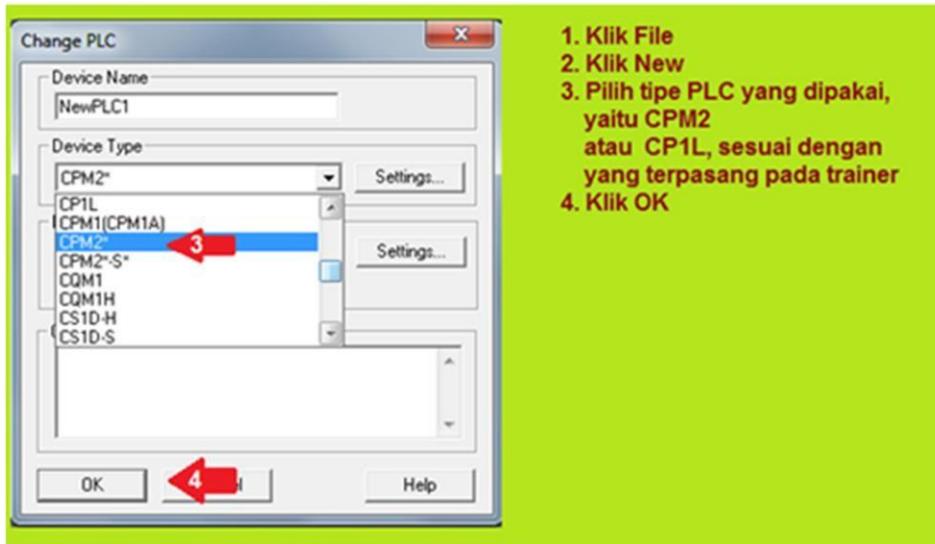
Q : membuat *Normal Close Coil*

6. Membuat Proyek Baru

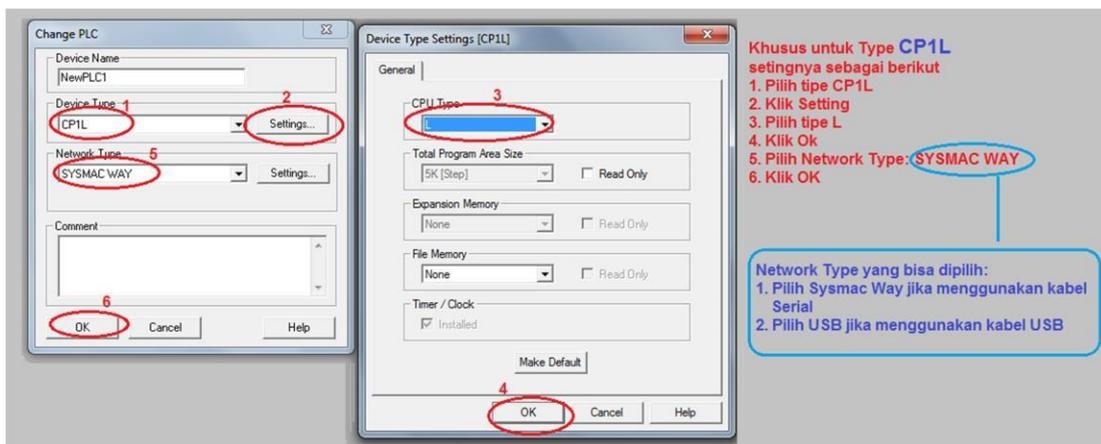
Klik menu File kemudian pilih New Setelah itu akan muncul kotak dialog pemilihan PLC seperti pada gambar



Gambar 1. 7 Kotak dialog pemilihan PLC



Gambar 1. 8 Langkah-langkah membuat proyek baru



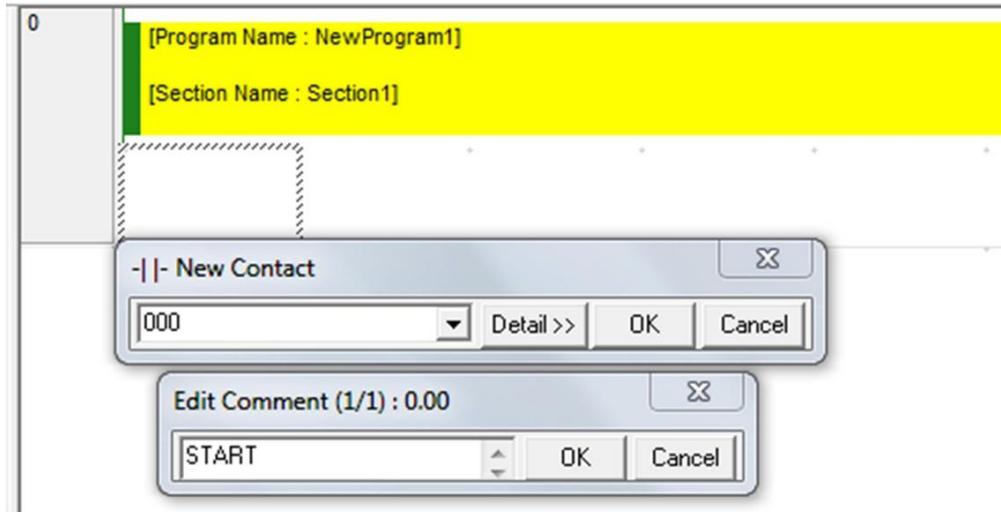
Gambar 1. 9 Penyetingan jenis PLC dan kabel komunikasi dari komputer ke PLC

untuk PLC Type CP1L/ CP1E

7. Proses Membuat *Ladder* Diagram (PLC Omron Tipe CP1E)

a. Membuat tombol *START*:

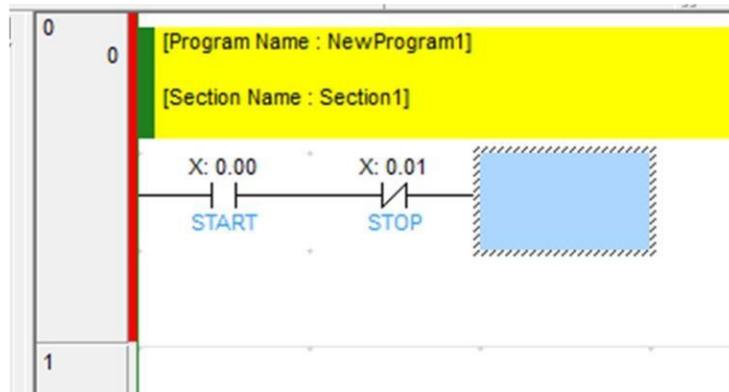
- Ketik C, => membuat kontak NO
- Setelah muncul kotak dialog *New Contack* => ketik *address* 000, klik OK
- Ketik *START*, kemudian klik OK.



Gambar 1. 10 Tampilan *Ladder* tombol *Start*

b. Membuat tombol *STOP*:

- Ketik / <= membuat kontak NC
- ketikan *adres* 001 , klik OK
- ketikan *STOP*, klik OK

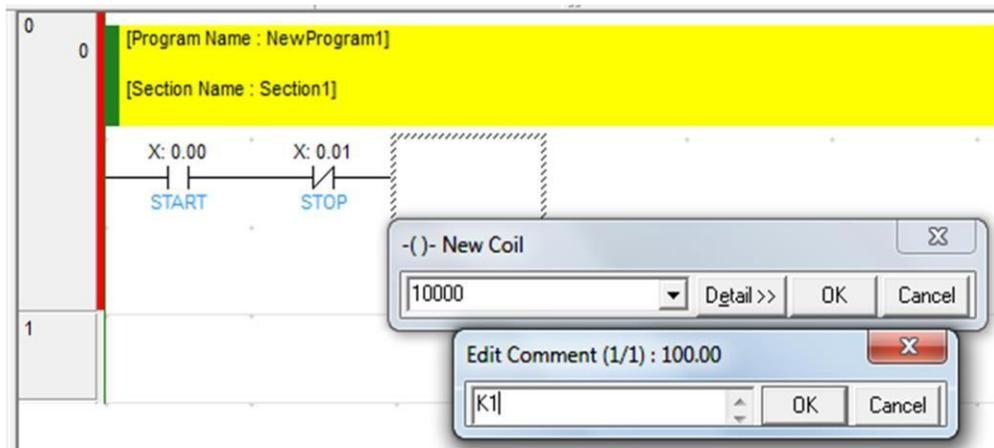


Gambar 1. 11 Tampilan *Ladder* tombol *stop*

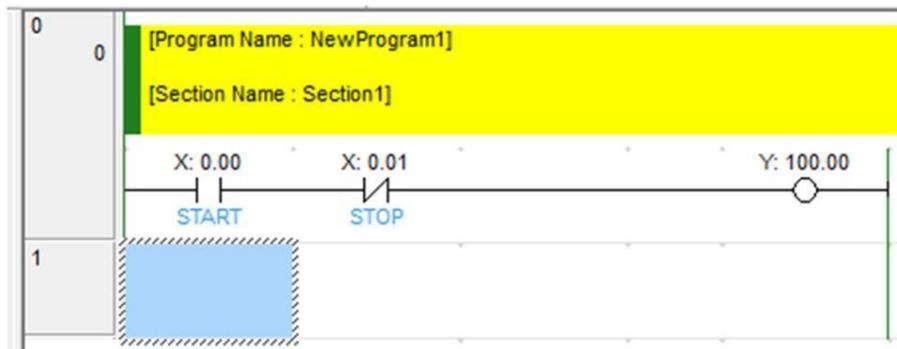
c. Membuat *Coil* K1:

- Ketik O <= membuat koil (*output*)
- Isikan *address* 10000, klik OK.
- Isikan komentar K1, klik OK

Maka akan dihasilkan satu baris *ladder* (RUNG)



Gambar 1. 12 Tampilan satu baris *ladder*

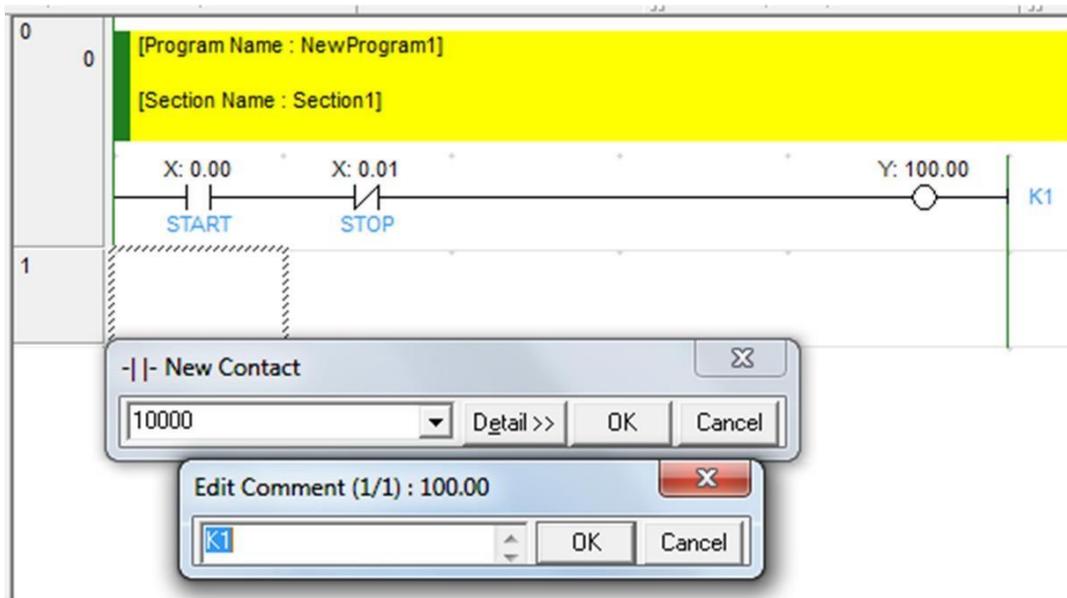


Gambar 1. 13 Tampilan *ladder*

d. Membuat pengunci:

- Klik pada tombol *START*, kemudian pindahkan kursor di bawah tombol *start*
- Ketikkan W <= membuat *Normally Open* OR
- Isikan 10000, klik OK

- Isikan K1, klik OK



Gambar 1. 14 Tampilan *ladder* membuat pengunci

- Simpan program tersebut dengan klik *File-Save* dan beri nama Rangkaian Pengunci Dasar.

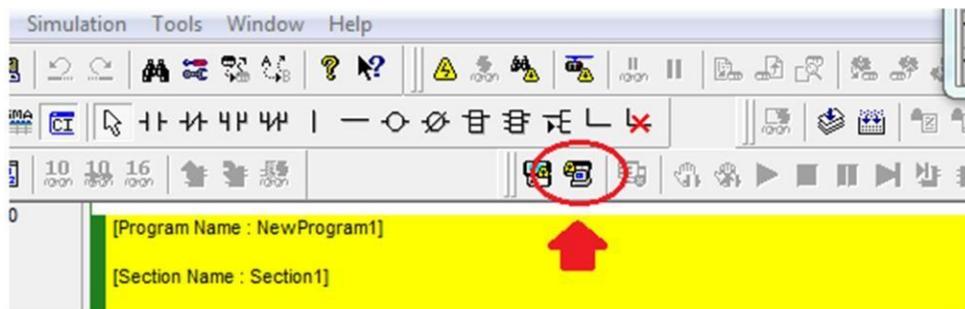
8. Program Dasar: Melakukan Simulasi

Program atau *Ladder* yang sudah dibuat, sebelum ditransfer ke PLC sebaiknya dilakukan uji coba pengetesan apakah program sudah berjalan sesuai dengan yang diinginkan atau belum.

Pengujian program dilakukan dengan cara disimulasikan. Pastikan bahwa pada komputer Anda telah terinstall software CX Programmer.

Full Version Cara melakukan simulasi adalah sebagai berikut:

- a. Pilih *type* PLC yang bisa digunakan untuk simulasi misalnya *type* : CJ1M, CP1L, CS1D-S dan lain-lain (pemilihan *type* plc ini dilakukan pada waktu akan mulai membuat program) karena tidak semua *Type* PLC bisa disimulasi.
- b. Setelah *Ladder* selesai dibuat dan disimpan, klik ikon *WORK ON LINE*



Gambar 1. 15 Tampilan *Toolbar*

- c. Tunggu proses *download* ke simulator. Proses selesai jika *Ladder* sudah ada yang berwarna hijau.



Gambar 1. 16 Tampilan *Ladder*

- d. Cara mensimulasikan mengoperasikan *input* (*Push Button* dll.) adalah:
 - Klik pada *input* yang akan dioperasikan
 - Menekan (menghidupkan) *switch*: Tekan tombol *keyboard* Ctrl + J
 - Melepas (mematikan) *switch* : Tekan tombol *keyboard* Ctrl + K

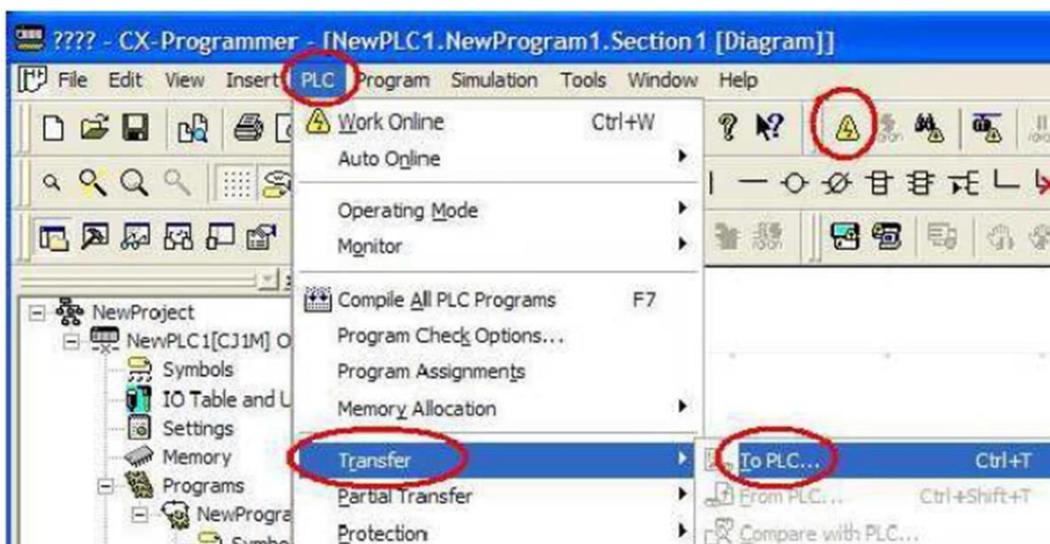
9. Program Dasar: Transfer Program ke PLC

Jika program (*ladder*) sudah diyakini benar, langkah berikutnya adalah melakukan transfer program ke PLC. Langkah persiapan: Pastikan bahwa

- program sudah benar dan disimpan ke komputer
- kabel data dari komputer ke PLC sudah terpasang
- PLC sudah terhubung ke *power supply* (sudah aktif)
- CX simulator tidak sedang aktif

Transfer program ke PLC:

- a. Klik ikon *Work Online*, atau tekan tombol *keyboard* Ctrl + W
- b. Klik menu PLC
- c. Pilih Transfer
- d. Pilih To PLC Tunggu beberapa saat, ikuti perintah/pesan yang muncul pada monitor



Gambar 1. 17 Langkah-langkah transfer program ke PLC

D. Tugas

- a. Sebutkan dan Jelaskan penerapan PLC pada dunia industri !

MODUL 2
MICROCONTROLLER



LABORATORIUM SISTEM OTOMASI
PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNIK
UNIVERSITAS KRISNADWIPAYANA
JAKARTA
2022

MODUL 2

MIKROKONTROLER

A. Tujuan Praktikum

1. Mempelajari cara memprogram Mikrokontroler.
2. Mencoba program sederhana untuk menampilkan karakter pada Lampu Mikrokontroler.

B. Alat Yang Digunakan

1. Laptop
2. Software Mikrokontroler
3. Mikrokontroller

C. Prosedur Praktikum

1. Praktikum dilakukan dengan menggunakan Mikrokontroler dan *software* Mikrokontroler untuk membuat program.

D. Pengolahan Data

1. Buatlah klasifikasi *input output* yang ada di Mikrokontroler.
2. Menjelaskan fungsi *Seven Segment common cathode* dan LCD yang ada di Mikrokontroler.
3. Menjelaskan kerja program asli dari Mikrokontroler.
4. Membuat dan menjelaskan program baru untuk Mikrokontroler.

E. Tinjauan Umum

1. Pengenalan Arduino

- Arduino Mega 2560



Gambar 2. 1 Arduino Trainer Kit

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

Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler.

SPEKIFIKASI

Mikrokontroler	: ATmega2560
Tegangan Operasional	: 5V
Tegangan <i>Input</i> (rekomendasi)	: 7-12V
Tegangan <i>Input (limit)</i>	: 6-20V
Pin Digital I/O	: 54 (of which 15 provide PWM output)
Pin Analog <i>Input</i>	:16
Arus DC per Pin I/O	: 20 mA
Arus DC untuk Pin 3.3 V	: 50 mA
Memori <i>Flash</i>	: 256 KB of which 8 KB used by <i>bootloader</i>
SRAM	: 8 KB
EEPROM	: 4 KB
<i>Clock Speed</i>	: 16 MHz
LED_BUILTIN	:13
Panjang	: 101.52 mm
Lebar	: 53.3 mm
Berat	: 37 g

2. Kekurangan Arduino Mega 2560 Rev 3 adalah sebagai berikut

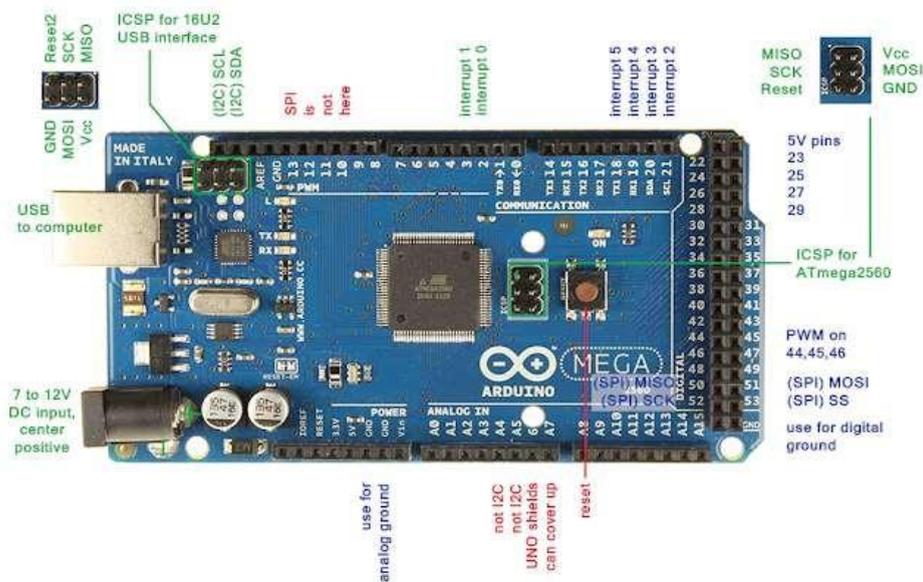
- Harga lebih mahal
- Ukuran lebih besar
- Bobot lebih berat

3. Keuntungan Arduino Mega 2560 adalah sebagai berikut:

- Memori SRAM (*Static Random Access Memory*) lebih besar 8 KB.
- Memori EEPROM (*Electrically Erasable Programmable Read Only Memory*) lebih besar 4 KB.
- Memori *Flash* (untuk diisi program) lebih besar, 256 KB, 8 KB digunakan *bootloader*.
- Jumlah pin lebih banyak.

4. Bagian-bagian Arduino

A. *Input/Output* Digital dan *output* analog



Gambar 2. 2 *Input* dan *Output* Digital

Input/output digital atau digital pin adalah pin-pin untuk menghubungkan arduino dengan komponen atau rangkaian digital. contohnya , jika ingin membuat LED berkedip, LED tersebut bisa dipasang pada salah satu pin *input* atau *output* digital dan *ground*.

komponen lain yang menghasilkan *output* digital atau menerima *input* digital bisa disambungkan ke pin-pin ini. *Input* analog atau analog pin adalah pin-pin yang berfungsi untuk menerima sinyal dari komponen atau rangkaian analog. contohnya ,potensiometer, sensor suhu, sensor cahaya, dll.

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai *input* atau *output*, menggunakan fungsi `pinMode()` , `digitalWrite()` , dan `digitalRead()`. Arduino Mega beroperasi pada tegangan 5 *volt*. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up* internal (yang terputus secara default) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- Serial : 0 (RX) dan 1 (TX); Serial 1 : 19 (RX) dan 18 (TX); Serial 2 : 17 (RX) dan 16 (TX); Serial 3 : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 Serial USB-to-TTL.
- Eksternal Interupsi : 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). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.
- SPI : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan

SPI. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.

- 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*).
- TWI : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan *Wire*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino Duemilanove atau Arduino Diecimila.

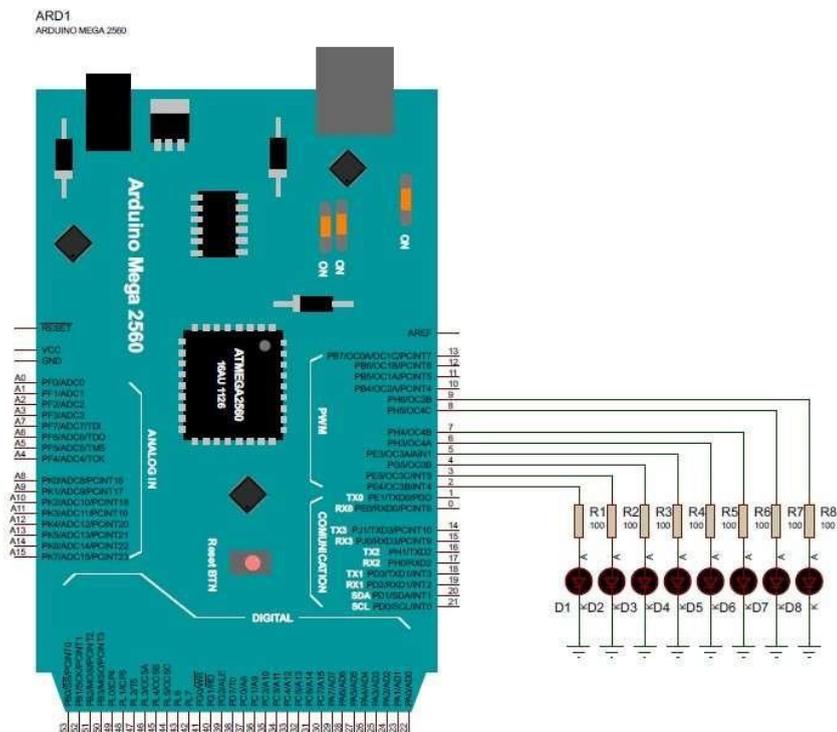
Arduino Mega2560 memiliki 16 pin sebagai analog *input*, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default* pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 *Volt*, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *analogReference()*.

Ada beberapa pin lainnya yang tersedia, antara lain:

- AREF : Referensi tegangan untuk *input* analog. Digunakan dengan fungsi *analogReference()*.
- 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.

5. Digital Output

Digital *output* adalah sebuah keluaran yang hanya memiliki 2 buah kondisi yaitu *LOW* dan *HIGH*. Untuk mengimplementasikan digital *output* kita memanfaatkan 16 buah led yang berada pada *trainer* Arduino. Led pada *trainer* ini ada 8 led yang dikonfigurasi sebagai *Led common cathode* (katode disambungkan semua dan dihubungkan dengan *ground*) serta ada 8 led yang dikonfigurasi sebagai *Led common anode* (anode disambungkan semua dan dihubungkan dengan +VCC). Maka untuk menyalakan led *common cathode* harus diberikan logika *HIGH* (5V), jika diberi logika *low* maka led mati karena kedua sisi mendapat sumber *ground*. Sedangkan untuk menyalakan led *common anode* harus diberikan logika *LOW* (0V), jika diberi logika *high* maka led mati karena kedua sisi mendapat sumber +VCC. Berikut ini gambar skematik dari led yang ada pada *trainer* Arduino.



Gambar 2. 3 skematik led Array *common cathode* pada arduino

- Program Membuat LED *Blink* Selama 1 Detik Menyala dan 1 Detik Mati.

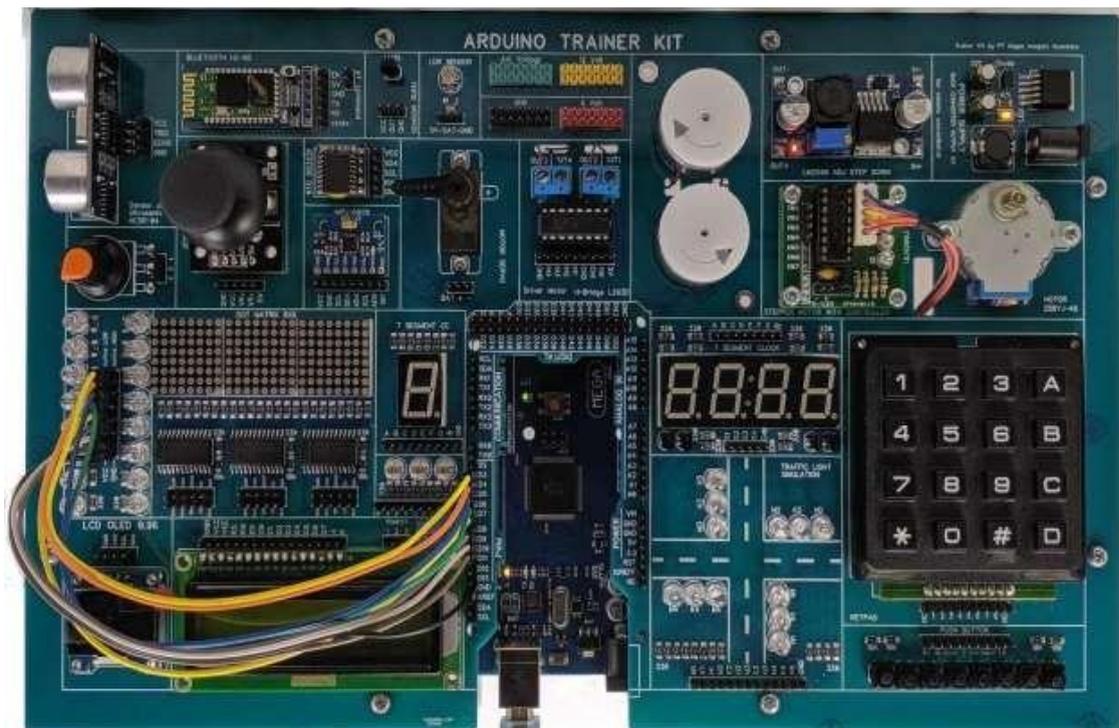
```
1 #define led1 2
2 #define led2 3
3 #define led3 4
4 #define led4 5
5 #define led5 6
6 #define led6 7
7 #define led7 8
8 #define led8 9
9
10 void setup() { pinMode(led1,
11     OUTPUT);
```

```
12 pinMode(led2, OUTPUT);
13 pinMode(led3, OUTPUT);
14 pinMode(led4, OUTPUT);
15 pinMode(led5, OUTPUT);
16 pinMode(led6, OUTPUT);
17 pinMode(led7, OUTPUT);
18 pinMode(led8, OUTPUT);
19 }
20
21 void loop() { digitalWrite(led1, HIGH);
22     digitalWrite(led2, HIGH);
23     digitalWrite(led3, HIGH);
24     digitalWrite(led4, HIGH);
25     digitalWrite(led5, HIGH);
26     digitalWrite(led6, HIGH);
27     digitalWrite(led7, HIGH);
28     digitalWrite(led8, HIGH);
29     delay(1000); digitalWrite(led1,
30     LOW); digitalWrite(led2, LOW);
31     digitalWrite(led3, LOW);
32     digitalWrite(led4, LOW);
33     digitalWrite(led5, LOW);
34     digitalWrite(led6, LOW);
35     digitalWrite(led7, LOW);
36     digitalWrite(led8, LOW);
37     delay(1000);
38 }
39
40
```

Penjelasan program

Baris	Keterangan
1-8	Deklarasi pin arduino yang digunakan, Proses deklarasi digunakan untuk mempermudah penulisan program, sehingga jika kita ingin merubah suatu pin menjadi pin yang lain kita hanyamerubah pin pada bagian deklarasi. #define led1 2 menunjukkan bahwa led1 adalah pin kaki arduino pin 2
11-18	Menginisialisasi bahwa led1 sampai led8 difungsikan sebagai <i>output</i> .
22-29	Memberi logika <i>high</i> atau 1 kepada semua pin led1 sampai pin led8, sehingga membuatsemua led menyala.
30	Fungsi menunda program selama 1 detik (1000 milidetik).
31-38	Memberi logika <i>low</i> atau 0 kepada semua pin led1 sampai pin led8, sehingga membuat semualed mati (karena led dikonfigurasi aktif <i>high</i>).
39	Fungsi menunda program selama 1 detik (1000 milidetik).

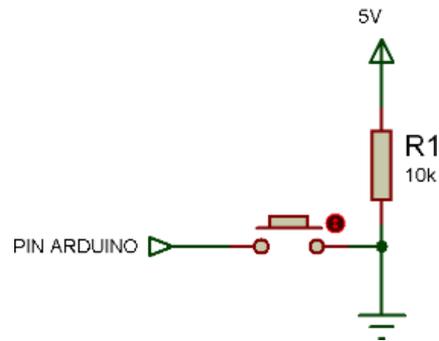
Program pada void loop() akan bekerja berulang sehingga led akan *blink* selama arduino menyala.



Gambar 2. 4 Wiring Led common cathode

6. Digital Input

Tombol merupakan bagian *input* dari sebuah *system* Arduino. Terdapat 2 jenis *input* digital tombol pada Arduino yaitu *Input Pullup External* dan *Input Pullup External*. Pada *pullup external* maka rangkaian tombol diharuskan menggunakan resistor *pullup* 10k seperti gambar dibawah ini:

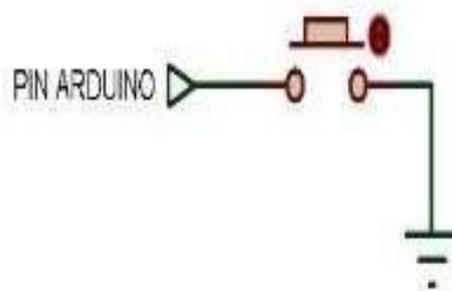


Gambar 2. 5 Skematik *Input* dengan *Pullup* Resistor

- **Program Inisialisasi *Input* dengan *Pullup External***

```
1 void setup() {  
2   pinMode(  
3     pin,  
4     INPUT);  
}
```

Sedangkan untuk *pullup* internal, rangkaian tidak perlu penambahan resistor *pullup*, karena sudah memanfaatkan *pullup* internal yang ada pada mikrokontroler. Berikut rangkaian *pullup* internal:



Gambar 2. 6 Skematik *Input* tanpa *Pullup* Resistor

- **Program Inisialisasi Input tanpa Pullup External**

```

1 void setup() {
2   pinMode(pin, INPUT_PULLUP);

```

Perlu penambahan *variable* tambahan untuk mempermudah dalam pembacaan data digital dari tombol, karena data digital tombol hanya 2 yaitu *LOW* atau *HIGH* maka digunakan data **Boolean**.

- **Program Pembacaan Tombol pada Arduino**

```

1 #define button1 A0 boolean
   data_button;
2
3
4 void setup() {
5   pinMode(button1, INPUT_PULLUP);
6   Serial.begin(9600);
7 }

```

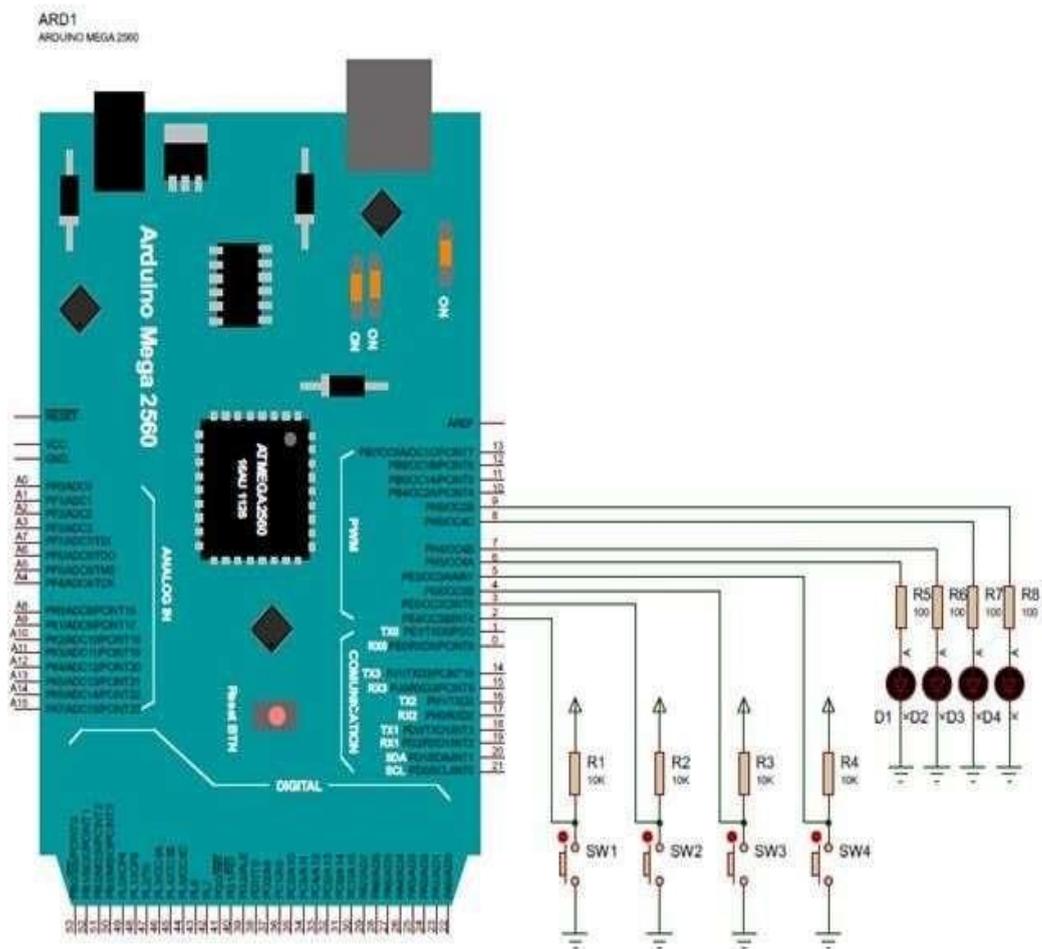
Penjelasan program:

Baris	Keterangan
1	Mendefinisikan sebuah nilai A0 menjadi button1. Hal ini dilakukan untuk mempermudah saat kita ingin merubah program dalam jumlah yang sangat banyak. #define menunjukkan bahwa kita mengetik button1 berarti itu adalah A0.
2	Membuat sebuah data Boolean dengan nama data_button Baris 5 menginisialisasi bahwa pada pin A0 (button1) digunakan sebagai input dengan mode pullup internal.

6	Mengaktifkan komunikasi serial dengan baudrate 9600.
10	Membaca nilai digital dari tombol.
11	Mencetak hasil dari data_button ke serial monitor.

Hasil program ini dapat dilihat pada serial monitor Arduino, jika tombol tidak ditekan maka hasil dari data_button adalah 1 dan jika tombol ditekan maka hasil dari data_button adalah 0. Hasil dari data_button inilah yang nantinya digunakan sebagai logika yang lebih lanjut.

Berikut ini gambar skematik yang ada pada *trainer* Arduino :



Gambar 2. 7 Skematik Pembacaan Tombol pada Arduino dengan Indikator LED

- **Program Membaca Tombol pada Arduino dengan Indikator LED**

```
1  #define button1 2
2  #define button2 3
3  #define button3 4
4  #define button4 5
5  #define led1 6
6  #define led2 7
7  #define led3 8
8  #define led4 9
9
10 void setup() { pinMode(button1, INPUT);
11
12     pinMode(button2, INPUT);
13     pinMode(button3, INPUT);
14     pinMode(button4, INPUT);
15     pinMode(led1, OUTPUT);
16     pinMode(led2, OUTPUT);
17     pinMode(led3, OUTPUT);
18     pinMode(led4, OUTPUT);
19 }
20
21 void loop() {
22     if(digitalRead(button1)==LOW){digitalWrite(led1, HIGH);}
23     else{ digitalWrite(led1, LOW);} if(digitalRead(button2)==LOW){digitalWrite(led2,
24     HIGH);}
25     else{ digitalWrite(led2, LOW);} if(digitalRead(button3)==LOW){digitalWrite(led3,
26     HIGH);}
27     else{ digitalWrite(led3, LOW);} if(digitalRead(button4)==LOW){digitalWrite(led4,
28     HIGH);}
```

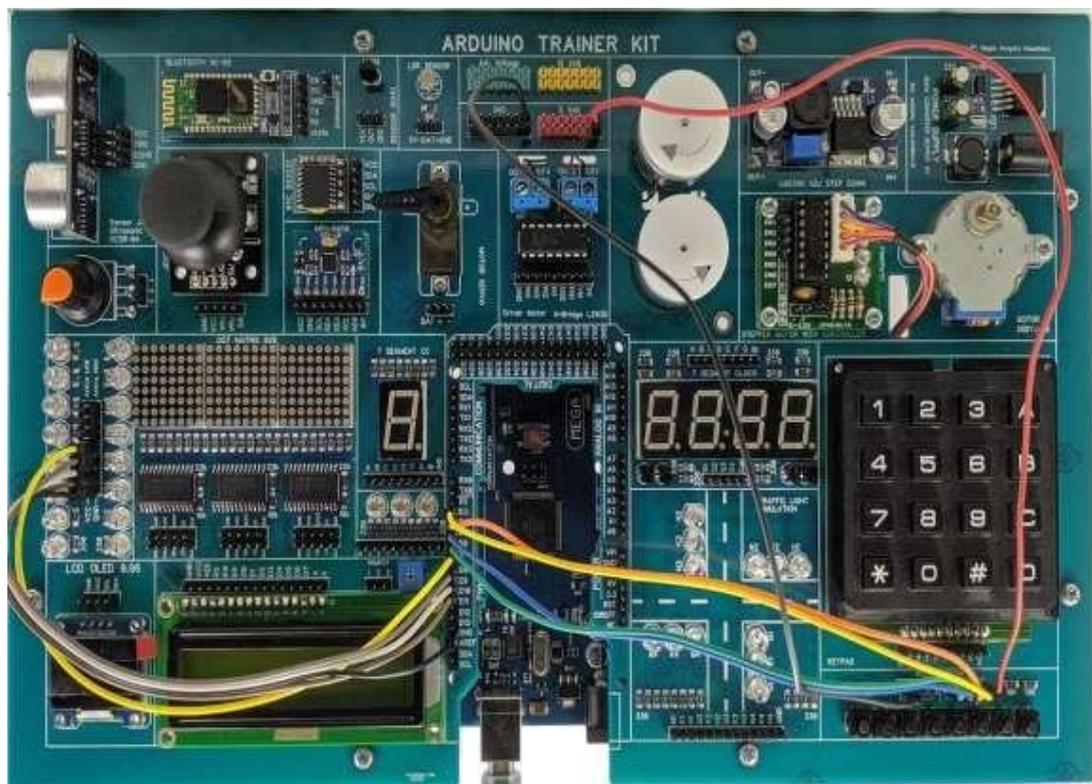
```

29 else{digitalWrite(led4, LOW);}
30 }
31
32
33
34

```

Penjelasan program:

Baris	Keterangan
1-8	<p>Deklarasi pin arduino yang digunakan, Proses deklarasi digunakan untuk mempermudah penulisan program, sehingga jika kita ingin merubah suatu pin menjadi pin yang lain kita hanya merubah pin pada bagian deklarasi.</p> <p>#define button1 2 menunjukkan bahwa button1 berada pada pin kaki arduino nomor 2</p> <p>#define led1 6 menunjukkan bahwa led1 berada pada pin kaki arduino nomor 6</p>
10-18	<p>Menginisialisasi bahwa button1 sampai button4 difungsikan sebagai input.</p> <p>Menginisialisasi bahwa led1 sampai led4 difungsikan sebagai output.</p>
21-33	<p>Suatu program perulangan dimana di dalamnya terdapat persyaratan yaitu apabila button1 s/d. button4 ditekan (berlogika <i>low</i>) maka led akan berlogika <i>high</i> sehingga akan menyala. Jika button1 s/d. button4 tidak ditekan (berlogika <i>high</i>) maka led akan berlogika <i>low</i> sehingga led dalam kondisi mati.</p>

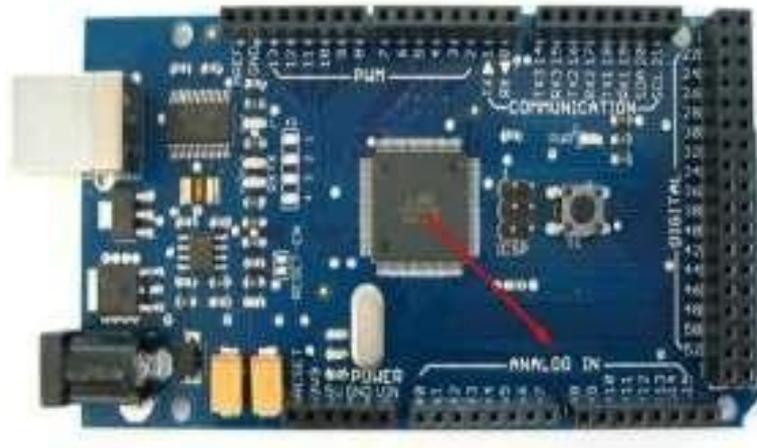


Gambar 2. 8 Wiring Led dan push button pada modul Arduino

7. Analog Input (Analog Read)

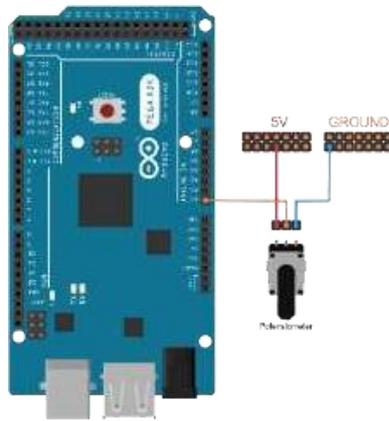
Analog input adalah sebuah masukan yang memiliki *range*, tidak seperti kita menggunakan tombol yang hanya dapat membaca *LOW* atau *HIGH*. Pada Arduino Mega *range* pembacaan analog *input* adalah 10-bit atau 0 – 1023. Artinya jika bernilai 0 maka *input* Arduino adalah 0 volt dan jika analog *Read* bernilai 1023 maka *input* ke Arduino adalah 5V. Biasanya *analog input* digunakan pada pembacaan sensor yang menggunakan data analog, seperti potensio, sensor suhu LM35, sensor cahaya, sensor api, dll. *Range* ini sangat penting dalam pemrograman karena nantinya nilai ini yang akan kita beri syarat sesuai dengan kebutuhan proyek yang akan dibuat. Misalnya jika sensor cahaya sudah mencapai nilai 500 maka lampu mati, tetapi jika sensor cahaya turun hingga nilai 200 maka lampu menyala.

Pin analog yang ada pada Arduino Mega adalah Pin A0 sampai A7.



Gambar 2. 9 Pin Analog pada Arduino Mega 2560

Berikut ini gambar diagram dari percobaan analog *read* menggunakan potensio:

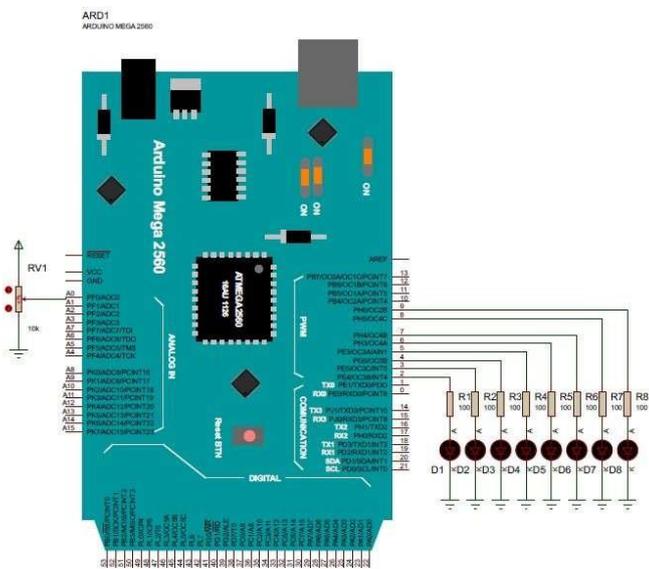


Gambar 2. 10 Wiring Diagram Potensiometer

- **Program Membaca data Analog dari Potensiometer**

1	<code>#define analogPin A0</code>
2	
3	<code>void setup() {</code>
4	<code>Serial.begin(9600);</code>
5	<code>}</code>
6	

Berikut ini gambar skematik yang ada pada trainer Arduino:



Gambar 2. 11 Membaca data Analog dari Potensiometer dengan Indikator LED

- **Program Membaca data Analog dari Potensiometer dengan Indikator LED**

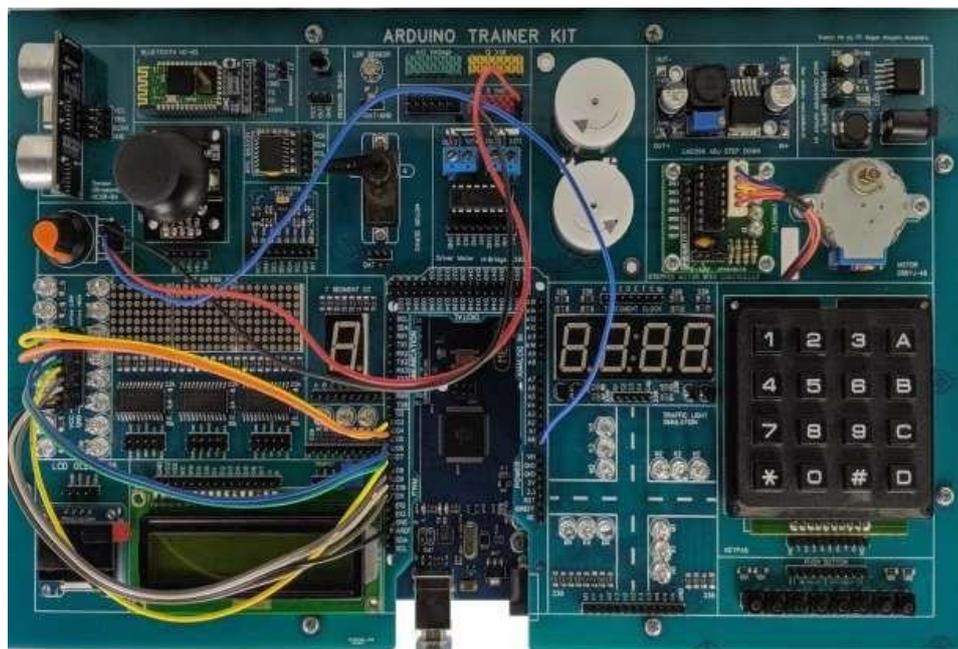
```

1  #define led1 2
2  #define led2 3
3  #define led3 4
4  #define led4 5
5  #define led5 6
6  #define led6 7
7  #define led7 8
8  #define led8 9
9  #define ADC_IN
10 A0
11
12 void setup() { pinMode(led1,
13   OUTPUT); pinMode(led2,
14   OUTPUT); pinMode(led3,
15   OUTPUT); pinMode(led4,
16   OUTPUT); pinMode(led5,
17   OUTPUT); pinMode(led6,
18   OUTPUT); pinMode(led7,
19   OUTPUT); pinMode(led8,
20   OUTPUT);
21   pinMode(ADC_IN,
22   INPUT);
23 }
24
25
26 void loop() {
27   if(analogRead(ADC_IN)<=24){ digitalWrite(led1, HIGH);} else{ digitalWrite(led1, LOW);}
28   if((analogRead(ADC_IN)>=25)&&(analogRead(ADC_IN)<=50)){ digitalWrite(led2,
29   HIGH);}
   else{ digitalWrite(led2, LOW);}
30   if((analogRead(ADC_IN)>=51)&&(analogRead(ADC_IN)<=75)){ digitalWrite(led3, HIGH);}
31   else{ digitalWrite(led3, LOW);}
32   if((analogRead(ADC_IN)>=76)&&(analogRead(ADC_IN)<=100)){ digitalWrite(led4,
33   HIGH);}
34   else{ digitalWrite(led4, LOW);}
35   if((analogRead(ADC_IN)>=101)&&(analogRead(ADC_IN)<=125)){ digitalWrite(led5,
36   HIGH);}
37   else{ digitalWrite(led5, LOW);}
38   if((analogRead(ADC_IN)>=126)&&(analogRead(ADC_IN)<=150)){ digitalWrite(led6,
39   HIGH);}
40   else{ digitalWrite(led6, LOW);}
41   if((analogRead(ADC_IN)>=151)&&(analogRead(ADC_IN)<=175)){ digitalWrite(led7,
42   HIGH);}
43   else{ digitalWrite(led7, LOW);}
44   if((analogRead(ADC_IN)>=176)&&(analogRead(ADC_IN)<=255)){ digitalWrite(led8,
45   HIGH);}
46   else{ digitalWrite(led8, LOW);}
47 }
48

```

Penjelasan program:

Baris	Keterangan
1-9	Deklarasi pin arduino yang digunakan, Proses deklarasi digunakan untuk mempermudah penulisan program, sehingga jika kita ingin merubah suatu pin menjadi pin yang lain kita hanya merubah pin pada bagian deklarasi. #define led1 2 menunjukkan bahwa led1 berada pada pin kaki arduino nomor 2 #define ADC_IN A0 menunjukkan bahwa ADC_IN berada pada pin analog arduino nomor A0
11-20	Menginisialisasi bahwa led1 sampai led8 difungsikan sebagai output. Menginisialisasi bahwa ADC_IN difungsikan sebagai input.
24-47	Suatu program perulangan dimana di dalamnya terdapat persyaratan yaitu apabila ADC_INterpenuhi dengan kondisi nilai tertentu, maka led1 sampai led8 akan menyala sesuai dengan nilai syarat ADC yang ditentukan. Jika tidak memenuhi kondisi syarat yang sudahditentukan maka led akan mati.



Gambar 2. 12 Wiring data Analog dari Potensiometer dengan Indikator LED

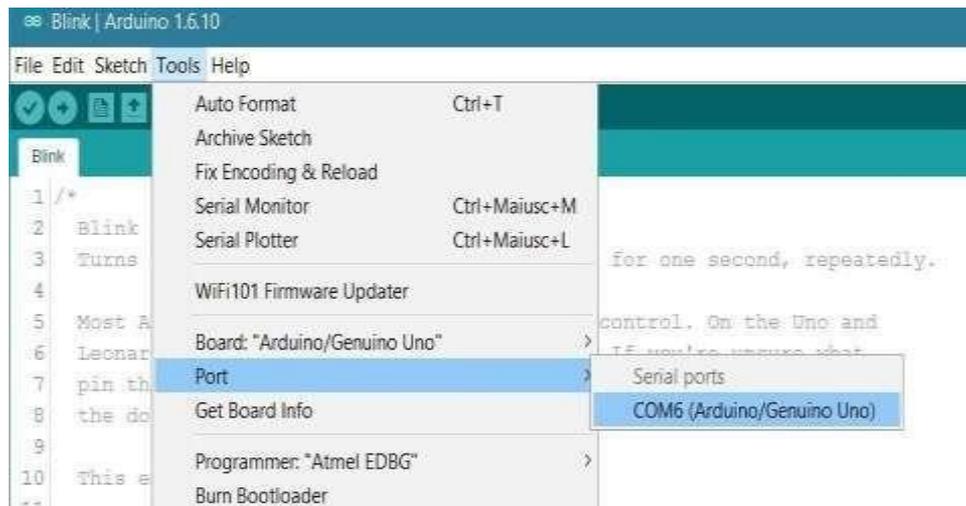
8. Software Arduino

a. Pengecekan driver CH340

Hubungkan Arduino Nano ke Komputer dengan menggunakan kabel mini USB untuk mengetahui instalasi driver CH340 berhasil atau tidak.

➤ Operating System Windows

- Buka aplikasi Arduino IDE
- Pilih *tools*
- Pilih *Ports* dan pastikan terdapat PORT yang terdeteksi (misal COM6)



Gambar 2. 13 Tampilan *software*

* Nomor COM dapat berubah menyesuaikan *port* komputer yang digunakan

➤ Operating System Windows

- Buka aplikasi Arduino IDE
- Pilih *tools*
- Pilih *Ports* dan pastikan terdapat *wchusbserial...* (misal `/dev/cu.wchusbserial1410`)



Gambar 2. 14 Tampilan *software*

b. Instalasi *Arduino basic library*

Lakukan tahap instalasi ICHIDUINO BASIC LIBRARY pada Arduino IDE dengan mengikuti langkah pada step dibawah ini:

1. Tutup aplikasi Arduino IDE
2. *Copy* file **ichiduinobasic.zip** ke folder document – Arduino –libraries
3. Klik kanan *extract here* pada file zip tersebut
4. Buka kembali Arduino IDE

c. Program *Line follower basic no index*

Saat pertama kali melakukan pemrograman, gunakan contoh library

Ichiduinobasic_no_index

Klik File – Example – Ichiduinobasic – Ichiduinobasic_no_index



Gambar 2. 15 Tampilan awal pemrograman



Gambar 2. 16 Tampilan program ichiduino basic tanpa index

Upload Program

- Pilih **Tools – Board – Arduino Nano**
- Pilih **Processor – Atmega328P ***
- Pilih **Tools – Port ****
- Klik **Upload**



Gambar 2. 17 Tampilan toolbar software

* Jika terjadi error saat upload program maka ganti **Arduino/Genuino Mega2560 (ATmega2560)**

** Nomor COM menyesuaikan laptop masing-masing

Jika terdapat teks **avrdude done. Thank you** maka program berhasil masuk ke robot.

F. Tugas

1. Sebutkan dan Jelaskan contoh penerapan seperti Arduino di bidang industri.