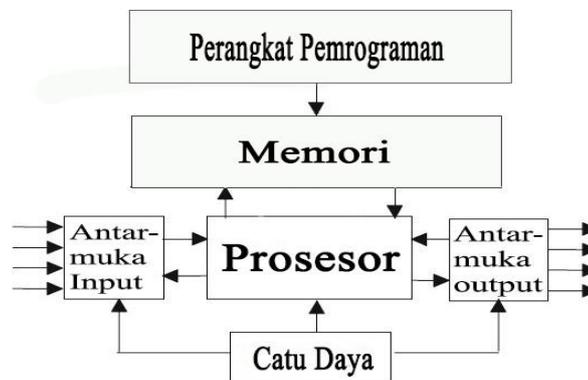


## BAB II

### TINJAUAN PUSTAKA

#### 2.1. *Programmable Logic Controller (PLC)*

PLC (*Programmable logic controller*) merupakan suatu bentuk khusus pengontrolan berbasis mikroprosesor yang memanfaatkan memori yang dapat diprogram untuk menyimpan intruksi-intruksi dan untuk mengimplementasikan fungsi – fungsi logika, sequencing, pewaktuan (*timing*), pencacahan (*counting*) dan aritmetika (*Programmable Logic Controller*) (Erlangga, 2003)



**Gambar 2.1** Sistem PLC (*Programmable Logic Controller*)  
(Erlangga, 2003)

PLC serupa dengan komputer namun, bedanya komputer dioptimalkan untuk tugas-tugas perhitungan dan penyajian data, sedangkan PLC dioptimalkan untuk tugas-tugas pengontrolan dan pengoperasian di dalam lingkungan industri. Dengan demikian PLC memiliki karakteristik :

1. Kokoh dan dirancang untuk tahan terhadap getaran, suhu, kelembaban, dan kebisingan.
2. Antarmuka input dan output telah tersedia secara built-in didalamnya.
3. Mudah diprogram dengan bahasa pemrograman yang mudah dipahami, yang sebagian besar berkaitan dengan operasi- operasi logika dan penyambungan.

### 2.1.1. Hardware

Umumnya, sebuah sistem PLC memiliki lima komponen dasar. Komponen-komponen ini adalah unit processor, memori, unit catu daya, bagian antarmuka input/output, dan perangkat pemrograman.

1. Unit processor atau central processing unit (unit pengolahan pusat) (CPU) adalah unit yang berisi mikroprocessor yang menginterpretasikan sinyal-sinyal input dan melaksanakan tindakan-tindakan pengontrolan, sesuai dengan program yang tersimpan di dalam memori, lalu mengkomunikasikan keputusan-keputusan yang diambilnya sebagai sinyal kontrol ke antarmuka output.
2. Unit catu daya diperlukan untuk mengkonversikan tegangan AC sumber menjadi tegangan rendah DC (5V) yang dibutuhkan oleh processor dan rangkaian-rangkaian di dalam modul-modul antarmuka input dan output.
3. Perangkat pemrograman dipergunakan untuk memasukkan program yang dibutuhkan ke dalam memori. Program tersebut dibuat dengan menggunakan perangkat ini dan kemudian dipindahkan ke dalam unit memori PLC.
4. Unit memori adalah tempat dimana program yang digunakan untuk melaksanakan tindakan-tindakan pengontrolan oleh mikroprocessor disimpan.
5. Bagian input dan output adalah antarmuka di mana prosesor menerima informasi dari dan mengkomunikasikan informasi kontrol ke perangkat-perangkat eksternal. Sinyal-sinyal input, oleh karenanya, dapat berasal dari saklar-saklar.

Tahap dasar untuk penyiapan awal untuk memudahkan dan memasukkan program dalam PLC dengan mempersiapkan daftar seluruh peralatan input dan output beserta lokasi I/O bit, penempatan lokasi word dalam penulisan data (*Programmable Logic Controller*) (Erlangga, 2003)

### 2.1.2. PLC Omron CP1E-E40 SDR-A

Merupakan Jenis dari PLC Omron seri CP1E, sedangkan arti dari E40 merupakan jumlah dari output dan input yang terdapat pada PLC. PLC jenis ini dapat di implementasikan pada penggerak mekanisme alat industri, alat rumah tangga, dan tugas teknik lainnya, yang mana bersifat logika elektronika (*Programmable Logic Controller*) (Erlangga, 2003)

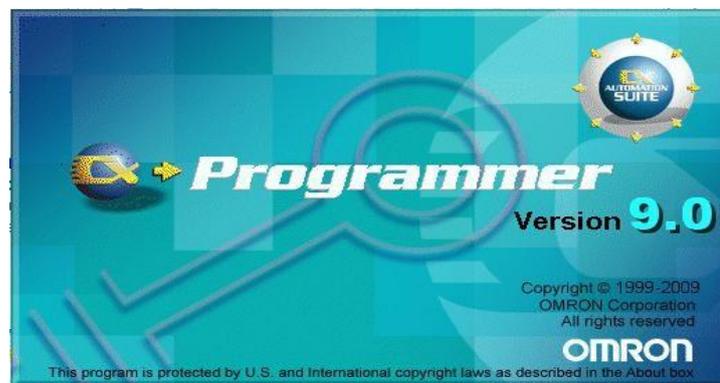


**Gambar 2.2** PLC Omron CP1E-E40 SDR-A  
(Erlangga, 2003)

PLC Omron seri CP1E memiliki I/O sebanyak 40 yang dimana 24 input bisa diubah menjadi analog, yaitu bekerja dengan tegangan 5 sampai 24 volt dan memiliki output sebanyak 16 yang dimana masing-masing output tersebut juga memiliki internal relay yang bekerja dengan arus hingga 10A. PLC Omron seri CP1E bekerja dengan tegangan yang bisa diubah 100 sampai 240 VAC, Program memory: 2Ksteps (EEPROM), Data memory DM: 2Kwords. Dan memiliki minimal tegangan kerja 5 VDC dan maksimum tegangan kerja 24 VDC pada input PLC. Kemudian pada masing-masing output PLC memiliki internal relay yang memiliki maksimum arus kerja sebesar 10 A. PLC Omron seri CP1E memiliki sistem program dengan menggunakan software pemrograman CX-Programmer (*Programmable Logic Controller*) (Erlangga, 2003)

### 2.1.3 Software CX-Programmer

**CX-Programmer** merupakan software khusus untuk memprogram PLC buatan OMRON. CX Programmer ini sendiri merupakan salah satu software bagian dari CX-One. Dengan CX-Programmer ini kita bisa memprogram aneka PLC buatan omron dan salah satu fitur yang saya suka yaitu adanya fitur simulasi tanpa harus terhubung dengan PLC, sehingga kita bisa mensimulasikan ladder yang kita buat, dan simulasi ini juga bisa kita hubungkan dengan HMI PLC Omron dibuat dengan menggunakan CX-Design (*Pengantar CX Programmer, 15*)



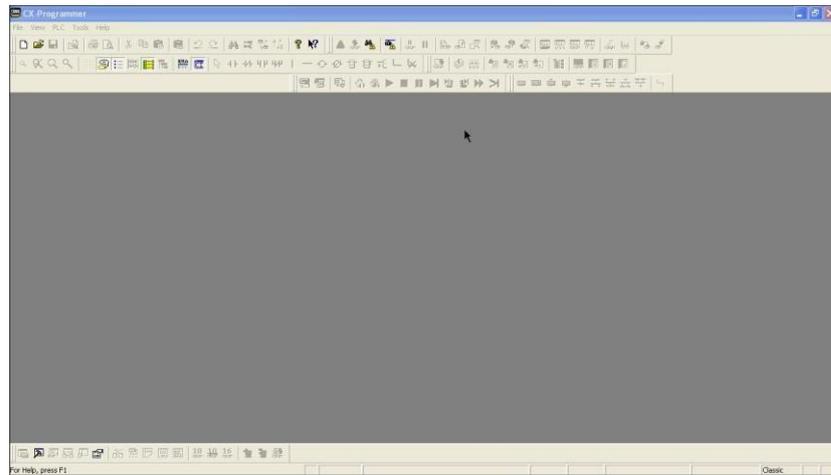
**Gambar 2.3** CX-Programmer Version 9.0 Omron

(*Pengantar CX Programmer, 15*)

Software ini beroperasi di bawah sistem operasi Windows, oleh sebab itu pemakai software ini diharapkan sudah familier dengan sistem operasi Windows antara lain untuk menjalankan software program aplikasi, membuat file, menyimpan file, mencetak file, menutup file, membuka file, dan keluar dari (menutup) software program. Ada beberapa persyaratan minimum yang harus dipenuhi untuk bisa mengoperasikan CX Programmer secara optimal yaitu: Komputer IBM PC/AT kompatibel

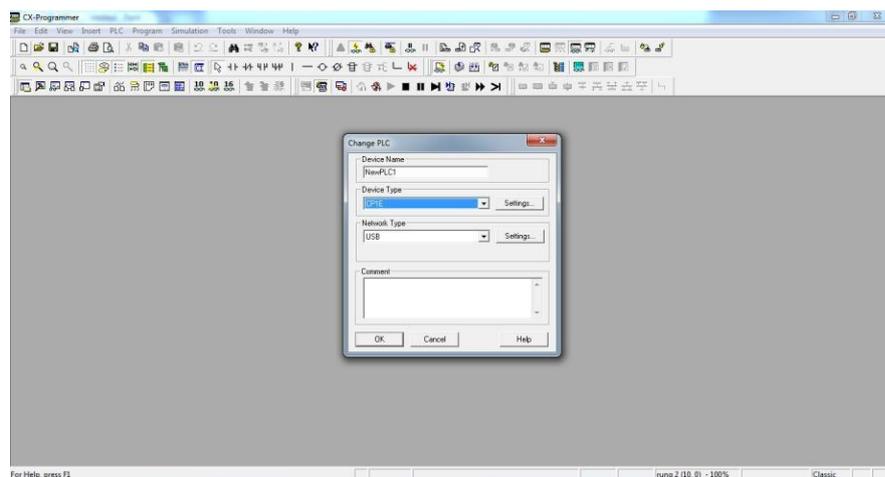
- CPU Pentium I minimal 133 MHz
- RAM 32 Mega bytes
- Hard disk dengan ruang kosong kurang lebih 100 MB
- Monitor SVGA dengan resolusi 800 x 600

Berikut tampilan dari CX-Programmer saat pertama kali dibuka :



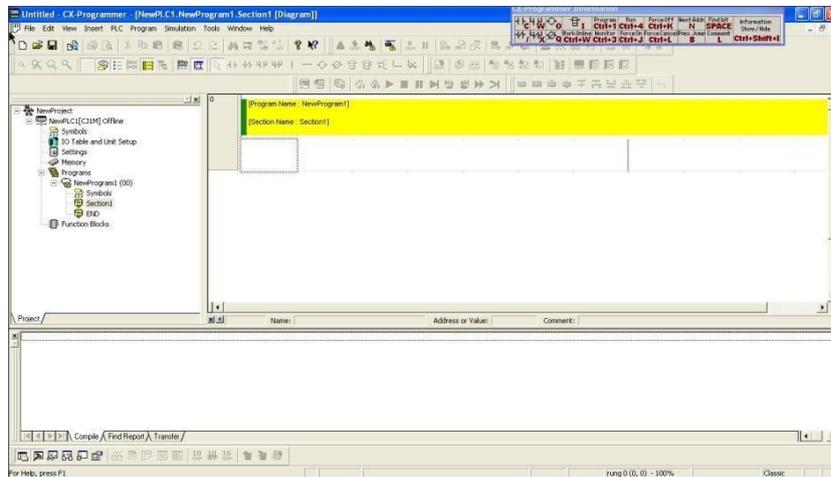
**Gambar 2.4** Tampilan Pertama Program CX-Programmer Version 9.0  
(*Pengantar CX Programmer, 15*)

Untuk memulai menggunakan CX-Programmer ini yaitu pada menu pilih file -> new atau bisa langsung pada toolbar klik gambar kertas putih untuk memulai membuat project baru, kalo untuk membuka file project yang sudah dibuat sebelumnya yaitu pilih file -> open atau pada toolbar pilih gambar disamping kertas putih maka akan muncul tampilan berikut :



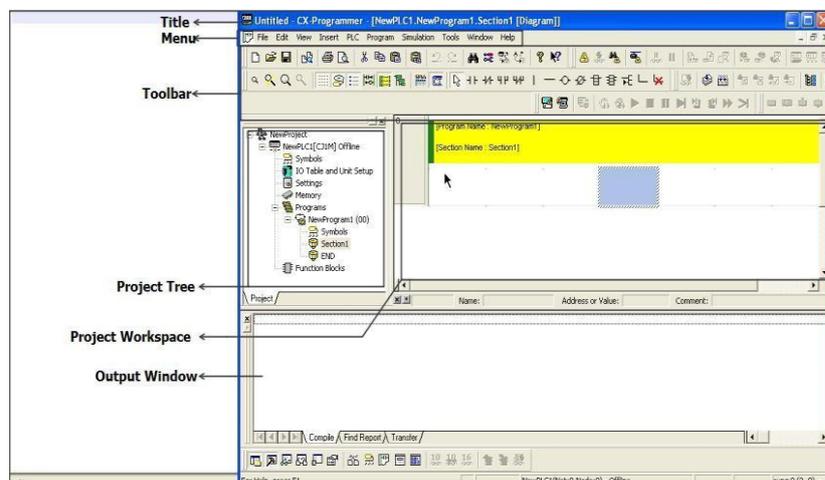
**Gambar 2.5** Tampilan Pemilihan Device PLC Pada Program  
(*Pengantar CX Programmer, 15*)

Setelah memilih tipe PLC yang akan digunakan, misalnya PLC CP1E dan Network type yang akan digunakan yaitu USB untuk setting lebih dalam bisa diklik setting, kemudian klik OK maka akan tampil tampilan berikut :



**Gambar 2.6** Tampilan Project Program CX-Programmer  
(*Pengantar CX Programmer, 15*)

Keterangan detail untuk tampilan tersebut yaitu:



**Gambar 2.7** Tampilan Keterangan Project Program CX-Programmer  
(*Pengantar CX Programmer, 15*)

### 2.1.4 Program PLC

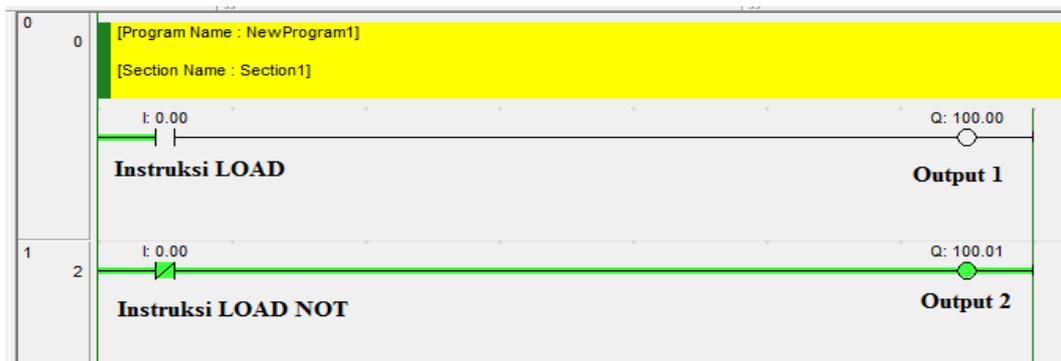
Ada 2 sistem pemrograman pada PLC Omron CP1E-E40SDR-A :

1. *Function Block Diagram* : Jenis Teknik Pemrograman Logic yang tersusun dari block-block diagram dalam 1 fungsi blok diagram khusus.
2. *Ladder Diagram* : Jenis Teknik Pemrograman Logic yang disusun dalam satuan-satuan kontak untuk menghasilkan fungsi tertentu dalam menghasilkan logika yang terdiri dari kontak NC, NO, Timer, Counter, dan lain-lain.

### 2.1.5 Instruksi Dasar Pada PLC

1. LOAD (LD) dan LOAD NOT (LD NOT)

Kondisi pertama yang mengawali sembarang blok logika di dalam diagram tangga berkaitan dengan instruksi LOAD (LD) atau LOAD NOT (LD NOT).



**Gambar 2.8** Ladder Diagram Instruksi LD dan LD NOT

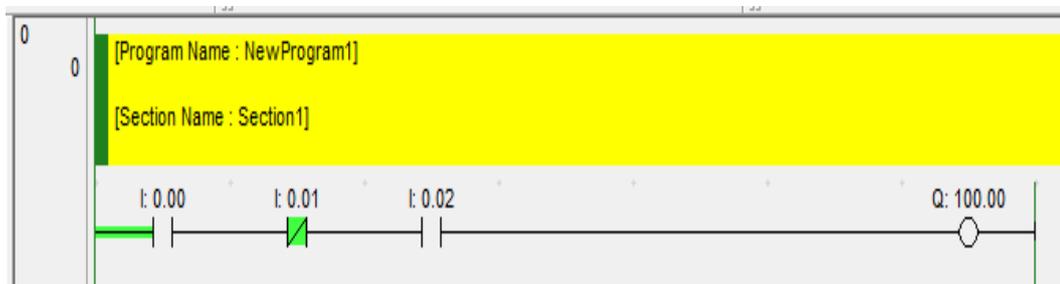
(Pengantar CX Programmer, 15)

**Tabel 2.1** Kode Mnemonic Instruksi LD dan LD NOT

Alamat	Instruksi	Operan
00000	LD	0.00
00001	OUT	100.00
00002	LD NOT	0.00
00003	OUT	100.01

## 2. AND dan AND NOT

Jika terdapat dua atau lebih kondisi yang dihubungkan seri pada garis instruksi yang sama maka kondisi pertama menggunakan instruksi LD atau LD NOT, dan sisanya menggunakan instruksi AND atau AND NOT. Gambar menunjukkan suatu penggalan diagram tangga yang mengandung tiga kondisi yang dihubungkan secara seri pada garis instruksi yang sama dan berkaitan dengan instruksi LD, AND NOT, dan AND. Masing-masing instruksi tersebut membutuhkan satu baris kode mnemonic (*Pengantar CX Programmer, 15*)



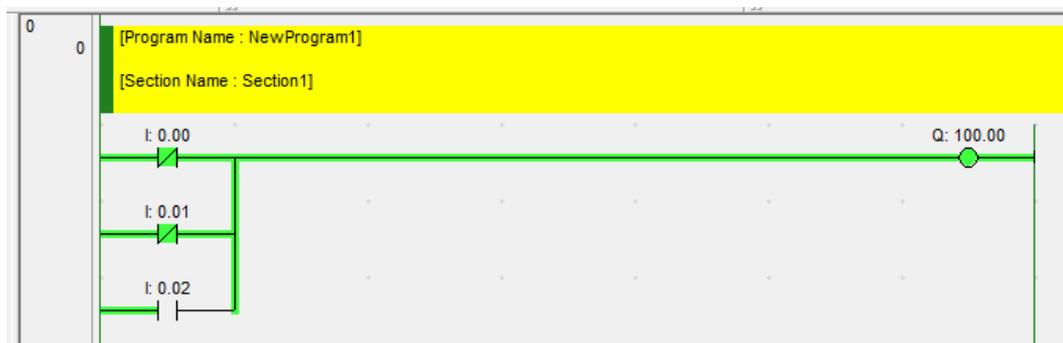
**Gambar 2.9** Ladder Diagram Instruksi AND dan AND NOT  
(*Pengantar CX Programmer, 15*)

**Tabel 2.2** Kode Mnemonic Instruksi AND dan AND NOT

Alamat	Instruksi	Operan
00000	LD	0.00
00001	AND NOT	0.01
00002	AND	0.02
00003	OUT	100.00

### 3. OR dan OR NOT

Jika dua atau lebih kondisi yang dihubungkan paralel, artinya dalam garis instruksi yang berbeda kemudian bergabung lagi dalam satu garis instruksi yang sama maka kondisi pertama terkait dengan instruksi LD dan LD NOT dan sisanya berkaitan dengan instruksi OR dan OR NOT. Gambar menunjukkan tiga buah instruksi yang berkaitan dengan instruksi LD NOT, OR NOT, dan OR. Masing-masing instruksi tersebut membutuhkan satu baris kode mnemonic (*Pengantar CX Programmer, 15*)



**Gambar 2.10** Ladder Diagram Instruksi OR dan OR NOT

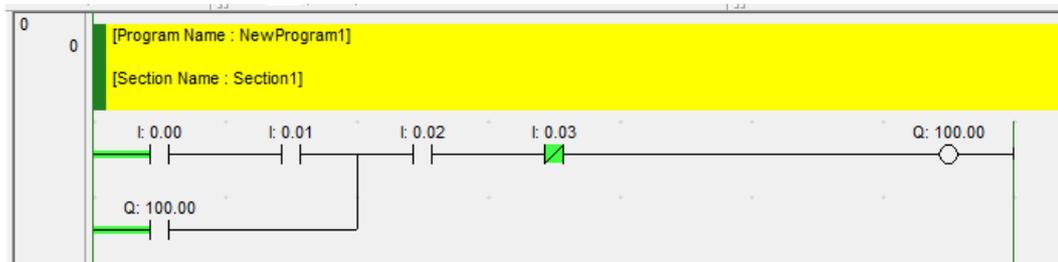
(*Pengantar CX Programmer, 15*)

**Tabel 2.3** Kode Mnemonic Instruksi OR dan OR NOT

Alamat	Instruksi	Operan
00000	LD NOT	0.00
00001	OR NOT	0.01
00002	OR	0.02
00003	OUT	100.00

#### 4. Kombinasi instruksi AND dan OR

Jika instruksi AND dan OR digabung atau dikombinasikan dalam suatu rangkaian tangga yang kompleks maka bisa dipandang satu persatu, artinya bisa dilihat masing-masing hasil gabungan dua kondisi menggunakan instruksi AND atau OR secara sendiri-sendiri kemudian menggabungkannya menjadi satu kondisi menggunakan instruksi AND atau OR yang terakhir. Gambar 2.11 menunjukkan contoh diagram tangga yang mengimplentasikan cara seperti tersebut di atas (*Pengantar CX Programmer, 15*)



**Gambar 2.11** Ladder Diagram Instruksi AND dan OR

(*Pengantar CX Programmer, 15*)

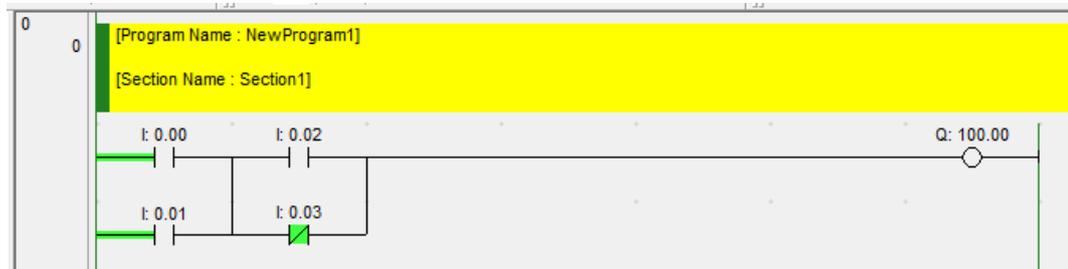
**Tabel 2.4** Kode Mnemonic Instruksi AND dan OR

Alamat	Instruksi	Operan
00000	LD	0.00
00001	AND	0.01
00002	OR	100.00
00003	AND	0.02
00004	AND NOT	0.03
00005	OUT	100.00

## 5. Instruksi-instruksi Blok Logika

Instruksi-instruksi blok logika tidak berhubungan dengan suatu kondisi tertentu pada diagram tangga, melainkan untuk menyatakan hubungan antar blok-blok logika, misalnya instruksi AND LD akan meng-AND-logik-kan kondisi eksekusi yang dihasilkan oleh dua blok logika, demikian juga dengan OR LD untuk meng-OR logikkan kondisi eksekusi yang dihasilkan dua blok logika (*Pengantar CX Programmer, 15*)

### a. AND LOAD (AND LD)



**Gambar 2.12** Ladder Diagram Instruksi AND LOAD (AND LD)

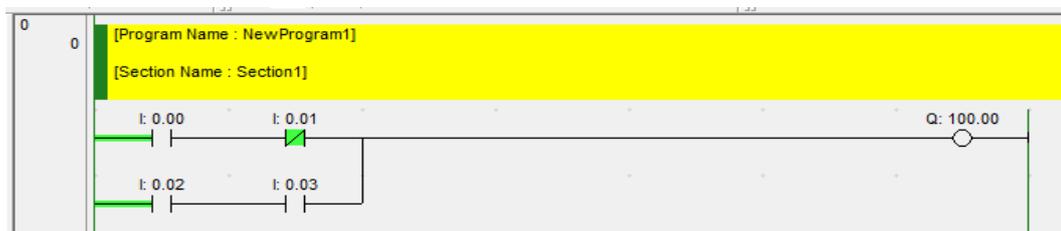
(*Pengantar CX Programmer, 15*)

**Tabel 2.5** Kode Mnemonic Instruksi AND LOAD (AND LD)

Alamat	Instruksi	Operan
00000	LD	0.00
00001	OR	0.01
00002	LD	0.02
00003	OR NOT	0.03
00004	AND LD	-
00005	OUT	100.00

b. OR LOAD (OR LD)

Instruksi ini digunakan untuk meng-OR-logik-kan dua blok logika. Gambar 2.13 menunjukkan contoh penggunaan blok logika OR LD yang terdiri atas dua blok logika. Kondisi eksekusi ON akan dihasilkan jika blok logik atas atau blok logika bawah dalam kondisi ON. Artinya, 0.00 dan kondisi ON dan 0.01 dalam kondisi OFF atau 0.02 dan 0.03 dalam kondisi ON) (*Pengantar CX Programmer, 15*)



**Gambar 2.13** Ladder Diagram Instruksi OR LOAD (OR LD)

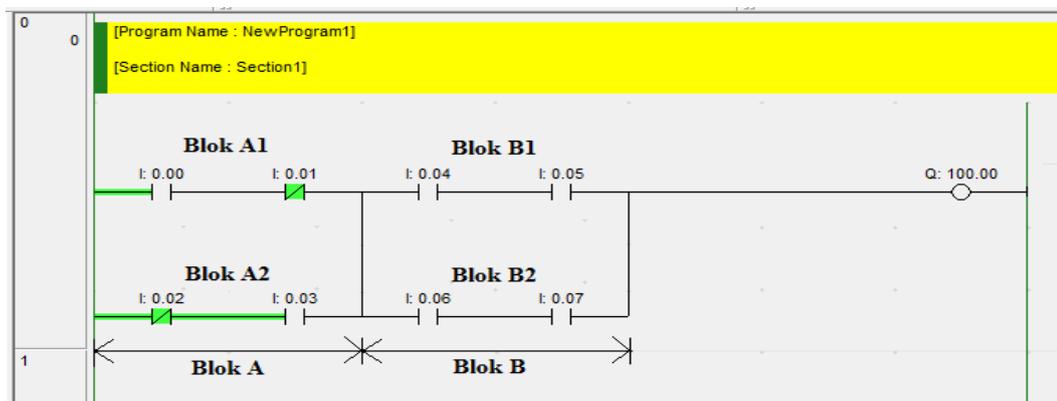
(*Pengantar CX Programmer, 15*)

**Tabel 2.6** Kode Mnemonic Instruksi OR LOAD (OR LD)

Alamat	Instruksi	Operan
00000	LD	0.00
00001	AND NOT	0.01
00002	LD	0.02
00003	AND	0.03
00004	OR LD	-
00005	OUT	100.00

c. Logika Kompleks

Untuk membuat kode mnemonik diagram tangga yang kompleks, caranya dengan cara membagi membagi diagram tersebut ke dalam blok- blok logika yang besar, kemudian membagi lagi blok yang besar tersebut menjadi blok- blok logika yang lebih kecil, demikian seterusnya hingga tidak perlu lagi dibuat blok yang lebih kecil lagi. Blok-blok ini kemudian masing-masing dikodekan, mulai dari yang kecil, dan digabungkan satu per satu hingga membentuk diagram tangga yang asli. Instruksi blok logika AND LD dan OR LD hanya digunakan untuk menggabungkan dua blok logika saja (blok logika yang digabungkan berupa hasil penggabungan sebelumnya, atau hanya sebuah kondisi tunggal). Gambar 2.13 memperlihatkan suatu contoh diagram tangga yang kompleks, yang dapat dibagi dua blok besar (blok A dan B). Blok A dapat dibagi lagi menjadi dua blok yang lebih kecil (blok A1 dan A2), dan blok B dibagi menjadi dua blok yang lebih kecil, yaitu blok B1 dan B2. Kemudian blok- blok logika yang kecil ini ditulis terlebih dahulu, diawali dengan menuliskan blok A1 (alamat 00000 dan 00001) dan blok A2 (alamat 00002 dan 00003), kemudian digabung menggunakan instruksi blok logik OR LD (alamat 00004). Selanjutnya blok B1 dituliskan (alamat 00005 dan 00006) dilanjutkan dengan blok B2 (alamat 00007 dan 00008) dan digabung dengan instruksi blok logik OR LD (alamat 00009). Hasilnya berupa blok A dan blok B yang kemudian juga digabung menggunakan blok logika AND LD (alamat 00010) (*Pengantar CX Programmer, 15*)



**Gambar 2.14** Penggunaan Instruksi Logika Kompleks

(*Pengantar CX Programmer, 15*)

**Tabel 2.7** Kode Mnemonic Contoh Penggunaan Instruksi Blok Logika Kompleks

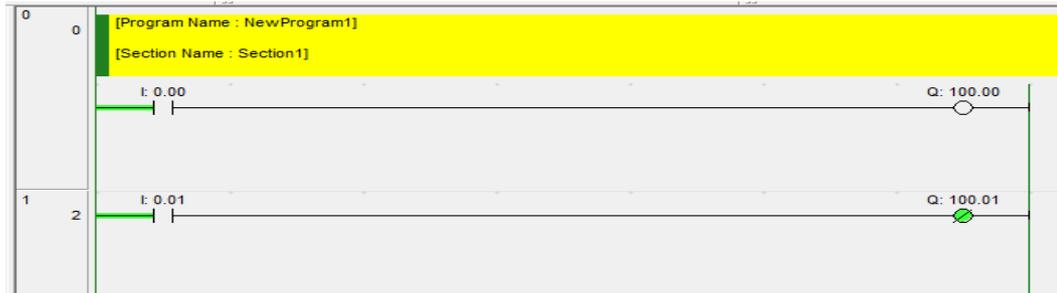
Alamat	Instruksi	Operan
00000	LD	0.00
00001	AND NOT	0.01
00002	LD NOT	0.02
00003	AND	0.03
00004	OR LD	-
00005	LD	0.04
00006	AND	0.05
00007	LD	0.06
00008	AND	0.07
00009	OR LD	-
00010	AND LD	-
00011	OUT	100.00

#### 6. Intruksi Kendali Bit

Terdapat instruksi dasar yang dapat digunakan untuk mengontrol status bit secara individual, yaitu DIFFERENTIATE UP (DIFU), DIFFERENTIATE DOWN (DIFD) instruksi ini dituliskan di sisi paling kanan diagram tangga dan membutuhkan sebuah alamat bit sebagai operan. Selain instruksi-instruksi ini digunakan untuk membuat bit-bit keluaran ON atau OFF dalam area IR (*Pengantar CX Programmer, 15*)

#### 7. Instruksi OUTPUT (OUT) dan OUTPUT NOT (OUT NOT)

Instruksi ini digunakan untuk mengontrol operan yang berkaitan dengan kondisiekseseksi (apakah ON atau OFF). Dengan menggunakan instruksi OUT, maka bit operan akan menjadi ON jika kondisi eksekusinya juga ON, sedangkan OUT NOT akan menyebabkan bit operan menjadi ON jika kondisi eksekusinya OFF (*Pengantar CX Programmer, 15*)



**Gambar 2.15** Ladder Diagram Instruksi LD OUT dan LD OUT NOT

(Pengantar CX Programmer, 15)

**Tabel 2.8** Kode Mnemonic Instruksi LD OUT

Alamat	Instruksi	Operan
00000	LD	0.00
00001	OUT	100.00

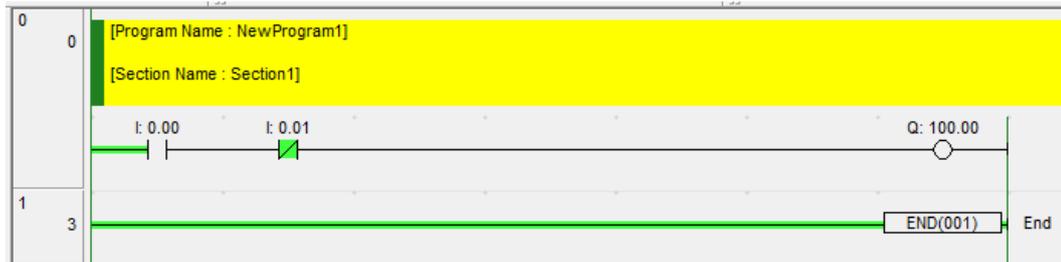
**Tabel 2.9** Kode Mnemonic Instruksi LD OUT NOT

Alamat	Instruksi	Operan
00000	LD	0.01
00001	OUT NOT	100.01

## 8. Instruksi END

Instruksi END merupakan instruksi terakhir yang harus dituliskan atau digambarkan dalam diagram tangga. CPU pada PLC akan mengerjakan semua instruksi dalam program dari awal (baris pertama) sampai ditemui instruksi END yang pertama, sebelum kembali lagi mengerjakan instruksi dalam program dari awal (artinya instruksi-instruksi yang ada di bawah instruksi END akan diabaikan). Instruksi END tidak memerlukan operan dan tidak boleh diawali dengan suatu kondisi seperti pada instruksi lainnya.

Suatu diagram tangga atau program PLC harus diakhiri dengan instruksi END, jika tidak maka program tidak dijalankan sama sekali. Angka yang dituliskan pada instruksi END pada kode mnemonik merupakan kode fungsinya. Gambar 2.18 memperlihatkan contoh penggunaan instruksi END (*Pengantar CX Programmer, 15*)



**Gambar 2.16** Ladder Diagram Instruksi END

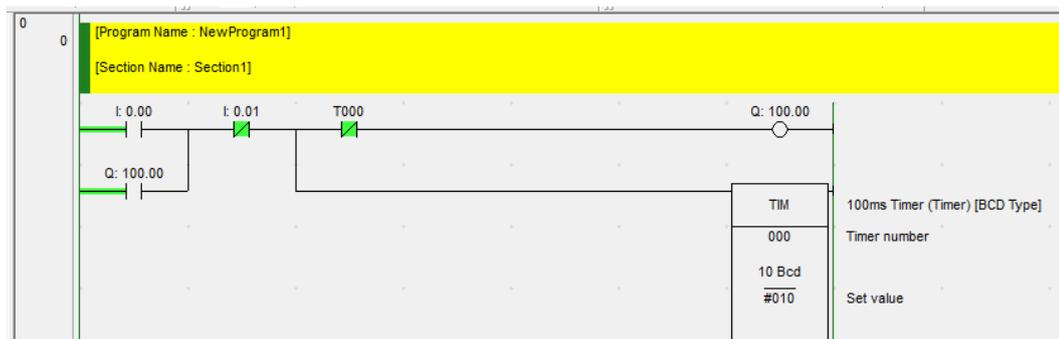
(*Pengantar CX Programmer, 15*)

**Tabel 2.10** Kode Mnemonic Instruksi END

Alamat	Instruksi	Operan
00000	LD	0.00
00001	AND NOT	0.01
00002	OUT	100.00
00003	END(001)	-

## 9. Instruksi TIMER (TIM)

Instruksi TIM dapat digunakan sebagai timer (pewaktu) ON-delay pada rangkaian relai. Gambar 2.18 memperlihatkan simbol tangga dan area data operan dari instruksi TIM. Instruksi TIM membutuhkan angka timer (N), dan nilai set (SV) antara 0000 sampai 9999 (artinya 000,0 sampai 999,9 detik) (*Pengantar CX Programmer, 15*)



**Gambar 2.17** Ladder Diagram Penggunaan Instruksi Blok Logika TIMER (TIM)

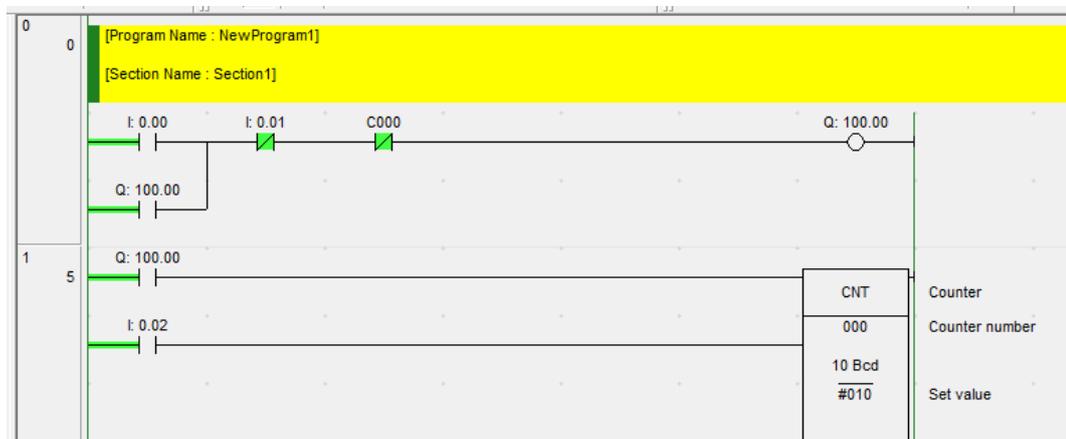
(*Pengantar CX Programmer, 15*)

**Tabel 2.11** Kode Mnemonic Instruksi Blok Logika TIMER (TIM)

Alamat	Instruksi	Operan
00000	LD	0.00
00001	OR	100.00
00002	AND NOT	0.01
00003	TIM	000 #010
00004	AND NOT	T000
00005	OUT	100.00

## 10. Instruksi COUNTER (CNT)

CNT yang digunakan di sini adalah counter penurunan yang diset awal. Penurunan satu hitungan setiap kali saat sebuah sinyal berubah dari OFF ke ON. Counter harus diprogram dengan input hitung, input reset, angka counter, dan nilai set (SV) Nilai set ini adalah 0000 sampai 9999. Gambar 2.20 memperlihatkan simbol tangga dan area data operan dari instruksi CNT, Dan Gambar 2.21 memperlihatkan contoh penggunaan instruksi END (*Pengantar CX Programmer, 15*)



**Gambar 2.18** Ladder Diagram Penggunaan Instruksi COUNTER (CNT)

(*Pengantar CX Programmer, 15*)

**Tabel 2.12** Kode Mnemonic Instruksi COUNTER (CNT)

Alamat	Instruksi	Operan
00000	LD	0.00
00001	OR	100.00
00002	AND NOT	0.01
00003	AND NOT	C000
00004	OUT	100.00
00005	LD	100.00
00006	LD	0.02
00006	CNT	000 #010

### **2.1.6 Perangkat – Perangkat Input**

Pada dasarnya PLC memerlukan sebuah input untuk melakukan perintah yang sudah diprogram dan di download ke PLC. Umumnya PLC hanya memerlukan input oleh sebuah saklar mekanis dan sensor. Sebuah saklar mekanis menghasilkan sinyal (atau sinyal-sinyal) hidup / mati sebagai akibat dari tertutup atau terbukanya saklar oleh suatu input mekanis. Dipasaran tersedia saklar-saklar dengan kontak normally open (normal terbuka) (NO) atau normally closed (normal tertutup) (NC) atau kontak yang dapat diatur sesuai kebutuhan dengan memilih kontak-kontak yang tepat (*Pengantar CX Programmer, 15*)

### **2.1.7 Perangkat-perangkat Output**

Port-port output sebuah PLC dapat berupa tipe relay atau tipe isolator-optik dengan transistor atau tipe triac, bergantung pada perangkat yang tersambung padanya, yang akan dikontrol. Secara umum, sinyal digital dari salah satu kanal output sebuah PLC digunakan untuk mengontrol sebuah actuator yang pada gilirannya mengontrol sebuah proses. Istilah actuator digunakan untuk perangkat yang mengubah sinyal listrik menjadi gerakan-gerakan mekanik yang kemudian digunakan untuk mengontrol proses. Berikut ini adalah beberapa contoh dari output PLC (*Pengantar CX Programmer, 15*)

## 2.2 Tombol Tekan

Prinsip kerja tombol tekan hampir sama dengan saklar tekan yang digunakan pada instalasi penerangan, bedanya jika saklar tekan jenis yang mempunyai togel akan langsung mengikat/mengunci, sedangkan pada tombol tekan tidak ada. Jadi tombol tekan setelah ditekan tidak akan mengunci, tetapi kembali keadaannya semula. Ada dua kontak yang dapat dilakukan oleh tombol tombol tekan, yaitu :

1. Kontak NO (Normally Open) Biasanya berwarna hijau.
2. Kontak NC (Normally Close) Biasanya berwarna Merah.



**Gambar 2.19** Tombol Kontak(*Pengantar CX Programmer, 15*)

## 2.3 Relay

Relay adalah suatu piranti yang bekerja berdasarkan elektromagnetik untuk menggerakkan sejumlah kontaktor (saklar) yang tersusun. Kontaktor akan tertutup (On) atau terbuka (Off) karena efek induksi magnet yang dihasilkan kumparan (induktor) ketika dialiri arus listrik. Berbeda dengan saklar dimana pergerakan kontaktor (On/Off) dilakukan manual tanpa perlu arus listrik (*Sensor Switches, 25*)



**Gambar 2.20** Relay

(*Pengantar CX Programmer, 15*)

## 2.4 Motor DC

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya memutar *impeller* pompa, *fan* atau *blower*, menggerakkan kompresor, mengangkat bahan dan lain-lain. Motor listrik digunakan juga di rumah (*mixer*, bor listrik, kipas angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor menggunakan sekitar 70% beban listrik total di industri.

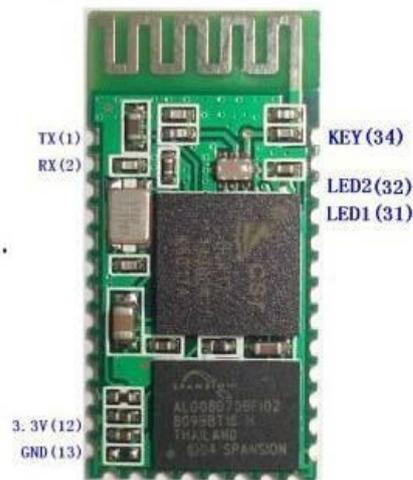
Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Untuk melihat bentuk serta komponen dari motor DC (*Teknik Tenaga Listrik dan Elektronika, 19*)

Motor DC memiliki 2 bagian dasar :

- Bagian yang tetap/stasioner yang disebut stator. Stator ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen.
- Bagian yang berputar disebut rotor. Rotor ini berupa sebuah koil dimana arus listrik mengalir.

## 2.5 Modul Bluetooth HC-05

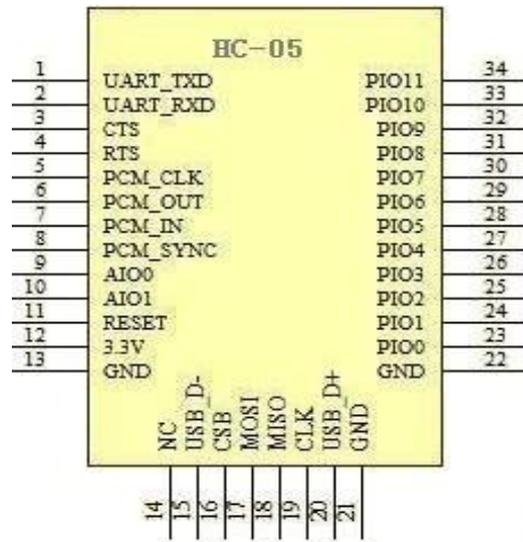
*Bluetooth* adalah protokol komunikasi *wireless* yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain<sup>1</sup>. Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe HC-05. modul *Bluetooth* HC-05 merupakan salah satu modul *Bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *Bluetooth* HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda – beda (*notebook HC-05-bluetooth, 37*)



**Gambar 2.21** Modul *Bluetooth* HC-05 (*notebook HC-05-bluetooth, 37*)

Modul *Bluetooth* HC-05 dengan *supply* tegangan sebesar 3,3 V ke pin 12 modul *Bluetooth* sebagai VCC. Pin 1 pada modul *Bluetooth* sebagai transmitter. kemudian pin 2 pada *Bluetooth* sebagai *receiver*.

Berikut merupakan konfigurasi pin *bluetooth* HC-05 ditunjukkan pada gambar dibawah ini:



**Gambar 2.22** Konfigurasi Pin HC-05 (*notebook HC-05-bluetooth, 37*)

Konfigurasi pin modul *Bluetooth* HC-05 dapat dilihat pada table 2.1 berikut ini :

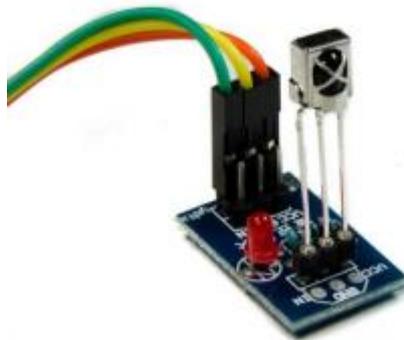
**Tabel 2.13** Konfigurasi pin *Module Bluetooth CH-05*

No.	Nomor Pin	Nama	Fungsi
1.	Pin 1	Key	-
2.	Pin 2	VCC	Sumber tegangan 5V
3.	Pin 3	GND	Groud tegangan
4.	Pin 4	TXD	Mengirim data
5.	Pin 5	RXD	Menerima data
6.	Pin 6	STATE	-

## 2.6 IR 1838

Komponen yang dapat menerima infra merah ini merupakan komponen yang peka cahaya yang dapat berupa dioda (fotodioda) atau transistor (fototransistor). Komponen ini akan merubah energi cahaya, dalam hal ini energi cahaya infra merah, menjadi pulsa-pulsa sinyal listrik. Komponen ini harus mampu mengumpulkan sinyal infra merah sebanyak mungkin sehingga pulsa-pulsa sinyal listrik yang dihasilkan kualitasnya cukup baik.

Semakin besar intensitas infra merah yang diterima maka sinyal pulsa listrik yang dihasilkan akan baik; jika sinyal infra merah yang diterima intensitasnya lemah, maka infra merah tersebut harus mempunyai pengumpul cahaya yang cukup baik dan sinyal pulsa yang dihasilkan oleh sensor infra merah ini harus dikuatkan.



**Gambar 2.23** Modul IR 1838 (*notebook IR 1838, 37*)

Pada prakteknya sinyal infra merah yang diterima intensitasnya sangat kecil sehingga perlu dikuatkan. Selain itu agar tidak terganggu oleh sinyal cahaya lain maka sinyal listrik yang dihasilkan oleh sensor infra merah harus disaring pada frekuensi sinyal pembawa, yaitu pada 30kHz-40kHz. Selanjutnya, baik fotodioda maupun fototransistor disebut sebagai fotodetector

## 2.7 Arduino

Arduino adalah sistem punarupa elektronika (electronic prototyping platform) berbasis open-source yang fleksibel dan mudah digunakan baik dari sisi perangkat keras/hardware maupun perangkat lunak/software. Di luar itu, kekuatan utama arduino adalah jumlah pemakai yang sangat banyak sehingga tersedia pustaka kode program (code library) maupun modul pendukung (hardware support modules) dalam jumlah yang sangat banyak. Hal ini memudahkan para pemula untuk mengenal dunia mikrokontroler.

Arduino didefinisikan sebagai sebuah platform elektronik yang open source, berbasis pada software dan hardware yang fleksibel dan mudah digunakan, yang ditujukan untuk seniman, desainer, hobbies dan setiap orang yang tertarik dalam membuat sebuah objek atau lingkungan yang interaktif (Sumber: Artanto,2012:1).

Arduino sebagai sebuah platform komputasi fisik (Physical Computing) yang open source pada board input output sederhana, yang dimaksud dengan platform komputasi fisik disini adalah sebuah sistem fisik yang interaktif dengan penggunaan software dan hardware yang dapat mendeteksi dan merespon situasi dan kondisi.

Kelebihan arduino dari platform hardware mikrokontroler lain adalah:

1. IDE Arduino merupakan multiplatform, yang dapat dijalankan di berbagai sistem operasi, seperti Windows, Macintosh dan Linux.
2. IDE Arduino dibuat berdasarkan pada IDE Processing sederhana sehingga mudah digunakan.
3. Pemrograman Arduino menggunakan kabel yang terhubung dengan port USB bukan port serial. Fitur ini berguna karena banyak komputer sekarang ini tidak memiliki port serial.
4. Arduino adalah hardware dan software open source, pembaca bisa mendownload software dan gambar rangkaian arduino tanpa harus membayar ke pembuat arduino. 5

5. Biaya hardware cukup murah, sehingga tidak terlalu menakutkan untuk membuat kesalahan.
6. Proyek arduino ini dikembangkan dalam lingkungan pendidikan sehingga bagi pemula akan lebih cepat dan mudah mempelajarinya.
7. Memiliki begitu banyak pengguna dan komunitas di internet dapat membantu setiap kesulitan yang dihadapi terutama oleh programmer pemula. (Sumber: Artanto,2012:2) 2.1.1

### **2.7.1 Sejarah Arduino**

Modul hardware Arduino diciptakan pertama kali di Ivrea, Italia pada tahun 2005 oleh Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David A. Mellis, dan Nicholas Zambetti (Arduino, 2011 dan Banzi 2008).

Bahasa Arduino merupakan fork (turunan) bahasa Wiring Platform dan bahasa Processing. Wiring Platform diciptakan oleh Hernando Barragan pada tahun 2003 (Wiring.org.co, 2011) dan Processing dibuat oleh Casey Reas dan Benjamin Fry pada tahun 2011 (Processing.org, 2012).

Arduino dikembangkan dari thesis Hernando Barragan di desain interaksi Institute Ivrea. Arduino dapat menerima masukan dari berbagai macam sensor dan juga dapat mengontrol lampu, motor, dan aktuator lainnya. Mikrokontroler pada board arduino di program menggunakan bahasa pemrograman arduino (based on wiring) dan IDE arduino (based on processing). Proyek arduino dapat berjalan sendiri atau juga bisa berkomunikasi dengan software yang berjalan pada komputer.

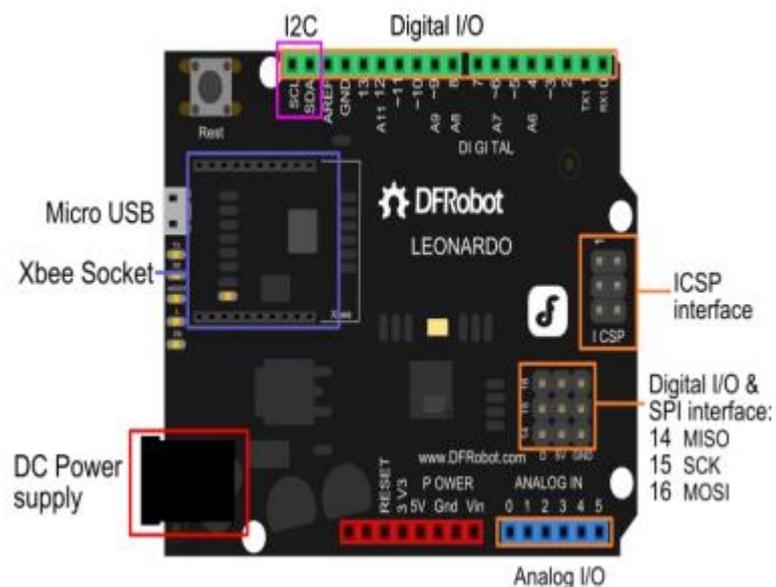
Arduino memakai standar lisensi open source, mencakup hardware (skema rangkaian, desain PCB), firmware bootloader, dokumen, serta perangkat lunak IDE (Integrated Development Environment) sebagai aplikasi programmer board Arduino. Setiap modul arduino menggunakan seri mikrokontroler yang berbeda seperti misalnya arduino leonardo yang menggunakan mikrokontroler ATmega328/32U4.

Menggunakan Arduino sangatlah membantu dalam membuat suatu prototyping ataupun untuk melakukan pembedaan proyek. Arduino memberikan 6 input output (I/O) yang sudah fix dan bisa digunakan dengan mudah. Arduino dapat digabungkan dengan modul elektro yang lain sehingga proses perakitan jauh lebih efisien.

Para desainer hanya tinggal membuat software untuk mendayagunakan rancang hardware yang ada. Software jauh lebih mudah untuk dimodifikasi tanpa harus memindahkan kabel. Saat ini arduino sangat mudah dijumpai dan ada beberapa perusahaan yang mengembangkan sistem hardware open source ini. (Sumber: Djuandi,2011:4).

### 2.7.2 Hardware Arduino

Papan arduino merupakan papan mikrokontroler yang berukuran kecil atau dapat diartikan juga dengan suatu rangkaian berukuran kecil yang didalamnya terdapat komputer berbentuk chip yang kecil. Pada Gambar 2.4 dibawah ini dapat dilihat sebuah papan arduino dengan beberapa bagian komponen didalamnya.(Sumber: Istiyanto, 2014:19)



**Gambar 2.24** arduino (notebook arduino, 18)

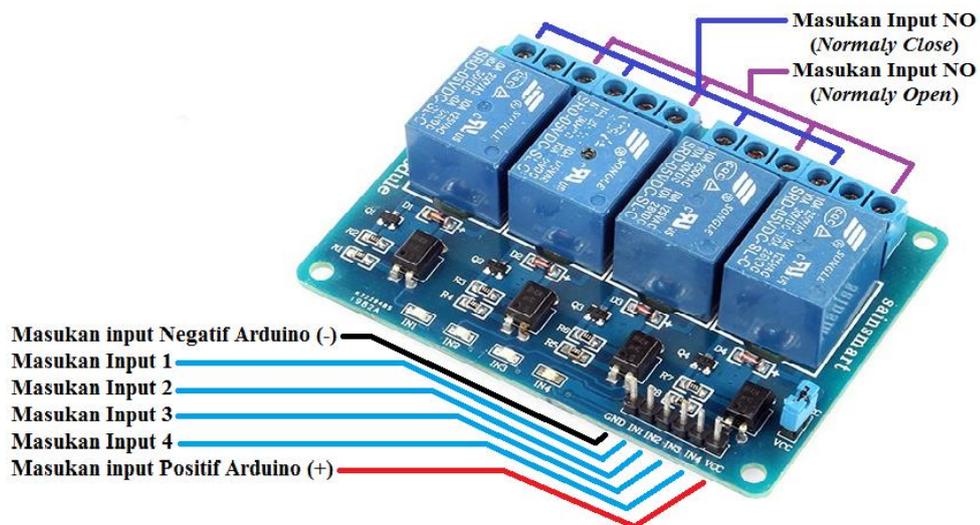
Pada hardware arduino terdapat 20 pin yang meliputi :

- a. 14 pin IO digital (pin 0-13) Sejumlah pin digital dengan nomor 0-13 yang dapat dijadikan input atau output yang diatur dengan cara program IDE
- b. 6 pin Input Analog (pin A0-A5) Sejumlah pin analog bernomor A0-A5 yang dapat digunakan untuk membaca nilai input yang memiliki nilai analog dan mengubahnya ke dalam angka antara 0 dan 1023.
- c. 6 pin Output analog (pin 3, 5, 6, 9, 10, dan 11) Sejumlah pin yang sebenarnya merupakan pin digital tetapi sejumlah pin tersebut dapat diprogram kembali menjadi pin output analog dengan cara membuatnya pada program IDE.

Papan arduino dapat mengambil daya dari USB port pada komputer dengan menggunakan USB charger atau dapat pula mengambil daya dengan menggunakan suatu AC adapter dengan tegangan 9 volt. Jika tidak terdapat power supply yang melalui AC adapter, maka papan arduino akan kembali mengambil daya dari USB port. Tetapi apabila diberikan daya melalui AC adapter secara bersamaan dengan USB port maka papan arduino akan mengambil daya melalui AC adapter secara otomatis. (Sumber: Istiyanto, 2014:19)

### 2.7.3 Rangkaian *Driver Relay*

Rangkaian *driver relay* berfungsi untuk mengendalikan motor arus searah (DC) yang dihasilkan dari *port* paralel I/O. Sinyal dari keluaran *port* biasanya berupa sinyal-sinyal yang kecil, sehingga tidak mampu untuk menggerakkan sistem daya berupa motor arus searah. Untuk dapat dimanfaatkan sinyal keluaran *port*, diperlukan suatu rangkaian *driver relay* agar sinyal yang kecil dapat dipergunakan untuk penggerak objek yang akan dikendalikan dari jarak jauh. Rangkaian *driver relay* ini dibangun oleh suatu komponen utama yaitu transistor dan *relay*. Transistor di rangkain *driver relay* difungsikan sebagai penguat sinyal dan *switching*, serta *relay* sebagai penggerak motor dc. *Driver relay* ini selain sebagai sebagai penguat dan *switching*, sekaligus difungsikan untuk mengendalikan motor dc dalam sistem pembalik putaran. Jadi, *driver relay* ini dapat mengatur arah putaran motor *forward* dan *reverse*. Semua *driver relay* pada sistem ini memiliki rangkaian dan karakteristik yang sama. Saat *relay* 1 bekerja maka posisi positif motor akan mendapat sumber tegangan positif dan posisi negatif motor terhubung dengan kutub negatif sumber tegangan. Sehingga, motor akan berputar dengan arah putaran searah jarum jam (*clockwise*). Dengan cara yang sama untuk menggerakkan kontak *relay* 2, maka terjadi kondisi yang berkebalikan yaitu motor akan berputar dengan arah putaran yang berlawanan arah jarum jam (*counter clockwise*).



**Gambar 2.25** *Driver Relay*

(channel-5v-relay-module-for-pic-arm-avr-dsp-arduino-msp430-ttl-logic.html. 2017)