**BAB II**

**TINJAUAN PUSTAKA DAN DASAR TEORI**

**2.1. Tinjauan Pustaka**

Saputra dkk (2011), melakukan penelitian dengan judul *“*Perancangan *Water Level Control* Menggunakan PLC *Omron Sysmac* C200H yang dilengkapi *Software SCADA Wonderware InTouch 10.5”* dalam penelitiannya ia membuat sebuah system otomasi *water level control* dikendalikan oleh PLC Omron *Sysmac* C200H dengan menggunakan panel *push button* yang terpasang pada *plant* atau juga dapat dikendalikan dan dimonitor melalui PC menggunakan *software* SCADA *Wonderware InTouch 10.5*. Selain itu sistem otomasi menggunakan SCADA ini dilengkapi *password* sebagai pengaman dari orang yang tidak bertanggungjawab. Pada penelitian ini telah dibuat suatu sistem otomasi *water level control* yang diaplikasikan pada *feed water tank* menggunakan PLC Omron *Sysmac* C200H yang dilengkapi dengan sistem SCADA (*Supervisory Control and Data Aqcuisition*) menggunakan *software Wonderware InTouch* versi 10.5.

Supriatna dkk (2010), melakukan penelitian dengan judul*“*Perancangan dan Implementasi Kontrol Level Air pada tangki Berbasis PLC (*Design and Implementation of* *Water Level Control at a Tank Based on PLC ”*Pada penelitian ini telah direalisasikan sebuah sistem pengontrolan tingkat *level* airmenggunakan PLC OMRON SYSMAC CP1H dengan input didapat dari sensor *level* air, *output analog* untuk mengatur besar bukaan *valve*, dan *output digital* untuk pengoperasian pompa. Parameter yang harus dipenuhi dalam sistem ini adalah kestabilan level air dan kestabilan debit air yang keluar dari tangki ke turbin. Untuk itu PLC digunakan untuk mengoptimalkan kerja dari *valve* dan pompa agar kedua parameter tersebut dapat terpenuhi dikarenakan keterbatasan antara logika *fuzzy* dan kemampuan motor untuk mengendalikan *valve* untuk menutup rapat.

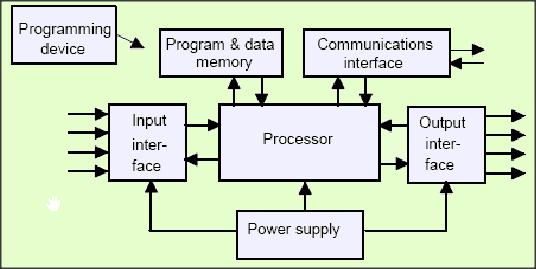
Syahreza (2010), melakukan penelitian dengan judul “Rancang Bangun Pengendali Otomatik Ketinggian *Fluida* dan Temperatur Menggunakan *Programmable Logic Controller* (*PLC*)” dalam penelitiannya ia merancang bangun suatu sistem pengendalian otomatik menggunakan PLC. Dalam sistem pengendalian ini, bahan pewarna (cair) sebagai bahan pertama dan air sebagai bahan kedua. Keduanya dialirkan dari masing-masing reservoir menggunakan pompa ke dalam tanki pencampuran dengan perbandingan *low* dan *high level.* Sistem ini akan mengatur ketinggian masing-masing bahan dalam tangki menggunakan sinyal digital. Dalam sistem pengendalian ini, PLC mempunyaiempat sinyal masukan digital, yaitu (1) tombol start danstop, (2) sensor *low level* ultrasonik, (3) sensor *high level*ultrasonik, dan (4) sensor temperatur. Sedangkan sinyalkeluaran digital PLC digunakan untuk menghidupkan dan mematikan kedua pompa, mengaduk bahan campuran, menghidupkan pemanas, dan membuka *solenoid valve*. Setelah tombol start *on*, sensor *level* ultrasonik belum mendeteksi adanya batas ketinggian bahan pewarna yang diinginkan, maka PLC melalui terminal output akan mengirimkan sinyal untuk mengalirkan bahan pewarna melalui pompa pertama hingga mencapai batas maksimum *low level* sensor - pompa pertama *off*. Kemudian dilanjutkan dengan mengirimkan sinyal kepompa kedua untuk mengalirkan air hingga mencapai batas ketinggian maksimum *high level* sensor. Sensor *low dan high level*ultrasonik digunakan sebagai pembatas akhir dari masukan bahan pertama dan bahan kedua. Tujuan dari rancang bangun sistem ini adalah mengembangkan sistem pengendali otomatik *plant* sederhana ketinggian fluida dan temperatur dalam satu tanki menggunakan PLC Omron. Pengendali Otomatik dibuat dengan mengembangkan program LLD, dengan metode mendesain menggunakan diagram STD yang akan mengendalikan sistem sesuai dengan keadaan yang diinginkan.

* 1. **Dasar Teori**
     1. ***Programmable Logic Controller* (PLC)**

*Programmable Logic Control* merupakan suatu pengontrol yang berbasis microprocessor dengan memanfaatkan memori yang dapat diprogram untuk menyimpan instruksi – instruksi dan untuk mengimplementasikan fungsi – fungsi seperti logika, sequencing, pewaktuan (*timing*), pencacahan (*counting*) dan aritmatika, untuk mengontrol berbagai jenis mesin serta proses melalui module I/O digital dan atau analog.

PLC menerima masukan dan menghasilkan keluaran sinyal-sinyal listrik untuk mengendalikan suatu *system*. Pada umumnya PLC dapat dibagi menjadi beberapa bagian, yaitu *central processing unit* (CPU), bagian *input/output,* memori, *power supply*, *programming device*.

1. Arsitektur PLC



Gambar 2.1 Arsitektur PLC.

* CPU (*Central Processing Unit*), yaitu otak dari PLC yang mengerjakan berbagai operasi, antara lain mengeksekusi program, menyimpan dan mengambil data dari memori, membaca kondisi/nilai *input* serta mengatur nilai *output*, memeriksa adanya kerusakan (*self - diagnosis*), serta melakukan komunikasi dengan perangkat lain. CPU berupa *Register,* ALU (*Arithmatic Logic Unit*) dan *Control unit.*
* *Input* merupakan bagian PLC yang berhubungan dengan perangkat luar yang memberikan masukan kepada CPU. Perangkat luar *input* dapat berupa tombol, *switch*, sensor atau piranti lain.
* *Output* merupakan bagian PLC yang berhubungan dengan perangkat luar yang memberikan keluaran dari CPU. Perangkat luar *output* dapat berupa lampu, katub (*valve*), motor dan perangkat – perangkat lain.
* Memori yaitu tempat untuk menyimpan program dan data yang akan dijalankan dan diolah oleh CPU. Dalam pembahasan PLC, memori sering disebut sebagai *file*. Dalam PLC memori terdiri atas memori program untuk menyimpan program yang akan dieksekusi, memori data untuk menyimpan nilai-nilai hasil operasi CPU, nilai *timer* dan *counter,* serta memori yang menyimpan nilai kondisi *input* dan *output*. Kebanyakan PLC sekarang memiliki satuan memori dalam *word* (16 bit). Memori berupa IC untuk menyimpan data dalam bilangan biner dan dapat berupa RAM, EPROM, EEPROM.
* Bus merupakan jalur informasi berupa data bus, *Address* dan *Control* bus.
* Fasilitas komunikasi, yang membantu CPU dalam melakukan pertukaran data dengan perangkat lain, termasuk juga berkomunikasi dengan komputer untuk melakukan pemrograman dan pemantauan.
* Fasilitas ekstensi, untuk menghubungkan modul PLC dengan modul pengembangan *input/output* sehingga jumlah terminal I/O dapat ditingkatkan.
* Catu daya, untuk memberikan sumber tegangan kepada semua komponen dalam PLC. Biasanya sumber tegangan PLC adalah 220 V AC atau 24 V DC.

1. Kelebihan dan kekurangan sistem PLC

Kelebihan PLC :

* Rangkaian kontrol dapat didesain ulang dengan sedikit atau tanpa perubahan pengawatan.
* Mampu mengontrol banyak alat dan melakukan perubahan dengan cepat.
* Memiliki dimensi fisik yang cukup kecil.
* Sinyal analog dan digital dapat dianalisis dan diproses.
* Dapat digunakan untuk sistem kendali yang terdistribusi.
* Mampu menghasilkan dokumentasi proses yang sedang dilakukan
* Mudah dalam menemukan kesalahan dan kerusakan.

Kekurangan PLC :

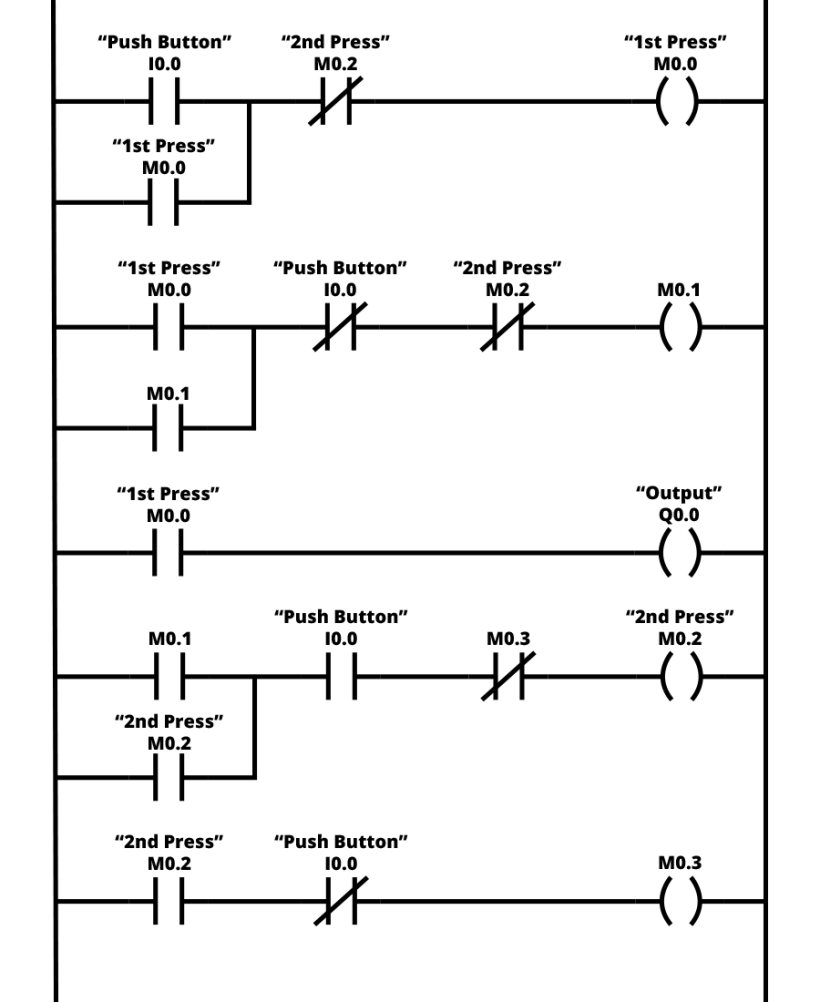
* Diperlukan kabinet agar alat kontrol elektronik dapat berada pada lingkungan yang terkontrol.
* Kabinet harus bersikulasi udara bersih dan temperatur harus dijaga
* Mahal untuk beberapa tipe system control.

1. Bahasa pemrograman *ladder diagram*

*Diagram ladder* (tangga) ialah bahasa pemrograman PLC dengan menggunakan simbol-simbol untuk menggambarkan kontak-kontak (*switches*) dan piranti-piranti keluaran (*output devices*) guna menggambarkan operasi suatu sistem. Penyajian berbentuk diagram (*graphical*) diinterpretasikan oleh piranti pemrograman ke dalam bahasa yang dapat di baca oleh PLC *processor*.

*Diagram ladder* mempunyai dua buah garis *vertical.* Terletak diantaranya dan menghubungkannya, berupa garis horisontal adalah aliran arus dan disebut juga *rungs* (anak tangga). Simbol-simbol yang menggambarkan operasi sirkuit disusun sesuai dengan urutan operasinya, yaitu piranti masukan (*input devices*) seperti switch dan sensor diletakkan di bagian kiri dan piranti keluaran untuk aktuator di bagian kanan *addres* atau alamat yang berupa angka-angka atau huruf atau gabungannya ditulis di atas setiap simbol.

Sebuah diagram tangga atau *ladder diagram* terdiri dari sebuah garis menurun ke bawah pada sisi kiri dengan garis-garis bercabang ke kanan. Garis yang ada di sebelah sisi kiri disebut sebagai palang bis (bus bar), sedangkan garis-garis cabang (*the branching lines*) adalah baris instruksi atau anak tangga. Sepanjang garis instruksi ditempatkan berbagai macam kondisi yang terhubungkan ke instruksi lain di sisi kanan. Kombinasi logika dari kondisi-kondisi tersebut menyatakan kapan dan bagaimana instruksi yang ada di sisi kanan tersebut dikerjakan.

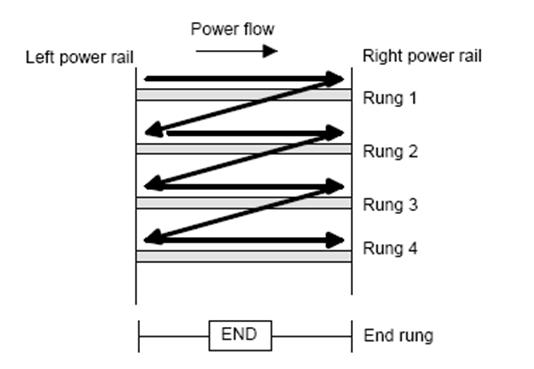


Gambar 2.2 *Ladder diagram.*

Sebagaimana ditunjukkan pada gambar 2.2 tersebut, sepanjang garis instruksi bisa bercabang-cabang lagi kemudian bergabung lagi. Garis-garis pasangan *vertikal* (seperti lambang kapasitor) itulah yang disebut kondisi. Pasangan garis *vertikal* yang tidak ada garis diagonalnya disebut sebagai Normal Terbuka – *Normally Open* atau NO serta terkait dengan instruksi *LOAD* (LD), *AND* atau OR. Sedangkan pasangan garis vertikal yang ada garis diagonal-nya dinamakan Normal Tertutup – *Normally Close* atau NC serta terkait dengan instruksi-instruksi *LD NOT, AND NOT* atau *OR NOT*.

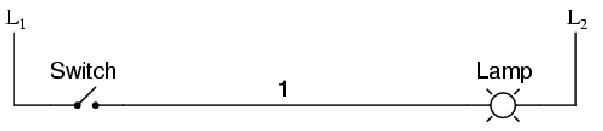
Aturan pemrograman dengan mempergunakan *ladder logic diagram* dapat dijabarkan sebagai berikut:

1. Dua garis *vertikal* pada *sheet* (= media untuk meletakkan komponen rangkaian) melambangkan daya. Di antara kedua garis tersebut komponen-komponen rangkaian dihubungkan sesuai dengan rancangan.
2. Masing-masing baris *ladder* mendefinisikan suatu operasi dalam proses kendali.
3. Masing-masing baris *ladder* wajib untuk dimulai dengan menempatkan sebuah *input* atau sejumlah input dan harus diakhiri dengan menempatkan sebuah *output*.
4. Perancangan *ladder* dengan menyesuaikan pada keadaan normal (*default*) perangkat listrik.
5. Suatu perangkat tertentu dapat digambarkan dengan menggunakan lebih dari satu buah baris/ *rung*.
6. Komponen-komponen input maupun output didefinisikan dengan menggunakan pengalamatan. Alamat tersebut merupakan indikasi dari lokasi komponen input maupun output dalam memori [PLC](http://ndoware.com/apa-itu-plc.html). Notasi masing-masing produk PLC berbeda-beda bergantung pada vendor yang memproduksinya.
7. Suatu keadaan komponen *output* dapat dipanggil sebagai keadaan komponen *input* dengan memanggil alamat komponen output yang diinginkan pada komponen input.
8. Pembacaan diagram dimulai dari kiri ke kana dan dari atas ke bawah seperti ditunjukkan pada gambar berikut:

[](http://ndoware.com/wp-content/uploads/2009/08/Arah-baca-ladder-logic-diagram.jpg)

Gambar 2.3 Arah baca diagram tangga.

Jika ingin membuat suatu *ladder diagram* yang sederhana untuk mengontrol sebuah lampu oleh sebuah saklar, maka bentuknya akan menjadi seperti berikut:

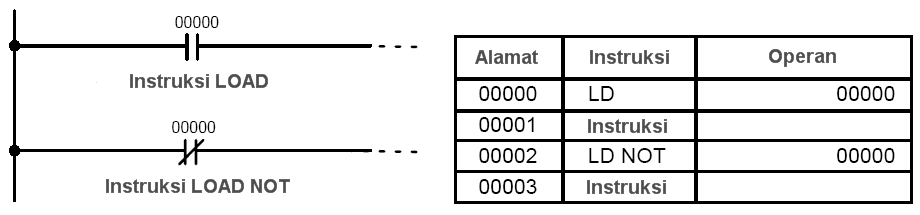


Gambar 2.4 Contoh *ladder diagram* sederhana.

“L1” dan “L2” menunjukkan pada dua *poles* dari sumber tegangan 120 VAC. Biasanya L1 adalah + 120 V dan L2 adalah *ground*. Terkadang pada PLC digunakan juga sumber tegangan selain 120 VAC, tetapi hal ini tidak akan mempengaruhi operasi dari *ladder diagram* ini.

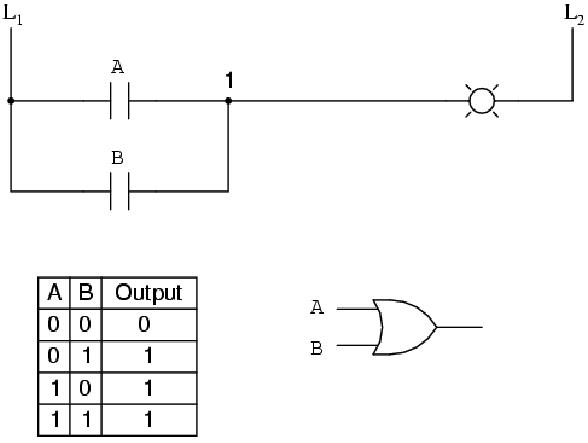
Pada pemrograman PLC dengan diagram *ladder* terdapat beberapa fungsi pemrograman yang dapat digunakan, fungsi-fungsinya yaitu:

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



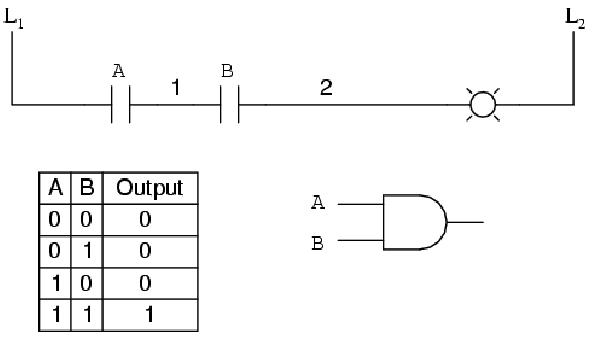
Gambar 2.5 Rangkaian LD dan LD NOT.

1. Fungsi OR



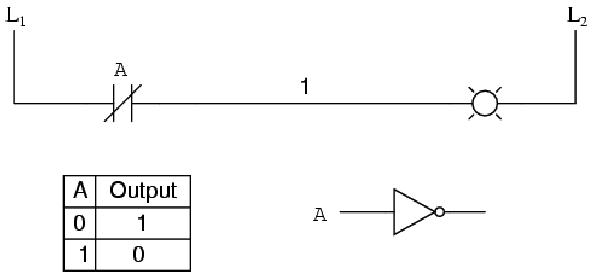
Gambar 2.6 Rangkaian OR.

1. Fungsi AND



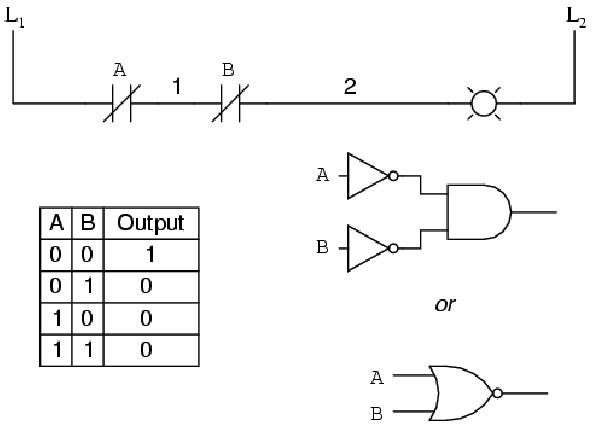
Gambar 2.7 Rangkaian AND.

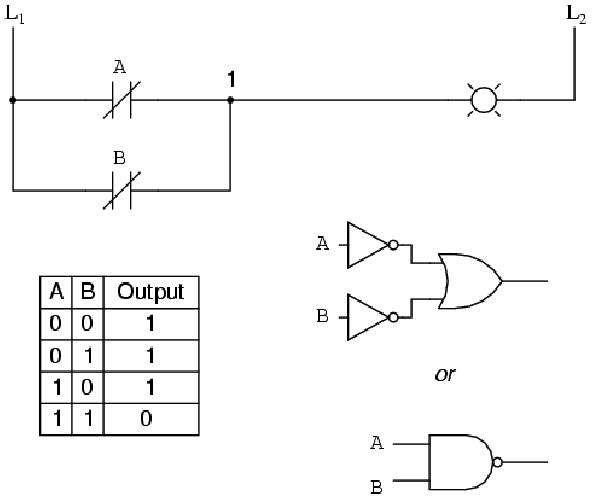
1. Fungsi Invers/ NOT



Gambar 2.8 Rangkaian NOT.

1. Fungsi OR NOT dan AND NOT





1. OR NOT (b) AND NOT

Gambar 2.9 Rangkaian (a) OR NOT ; (b) AND NOT.

1. Instruksi-instruksi umum pada *Pemrograman Ladder Logic*

Simbol yang digunakan pada pemrograman ladder logic dapat dibagi menjadi dua kategori yaitu *contact* (*input*) dan *coil*(*output*).

* *Contacts* (*input*),Kontak atau input pada pemrograman ladder logic terbagi menjadi dua yaitu kontak *Normally Open* yang biasa disebut instruksi *Examine On* atau *Examine If Close* (XIC) dan kontak *Normally Close* yang biasa disebut instruksi *Examine Off* atau *Examine If Open* (XIO).
* Kontak *Normally Open* (XIC), Merupakan suatu instruksi yang memberi informasi kepada prosesor untuk mencari kondisi ON pada alamat simbol. Jika kondisi ON terjadi, instruksi yang disampaikan berlogika benar (*true*). Atau lebih mudah dikatakan jika kontak tertutup, maka arus dapat mengalir.



Gambar 2.10 Simbol kontak normally open (XIC).

* Kontak *Normally Close* (XIO), memberi informasi ke prosesor untuk sebuah kondisi OFF. Alamat sebenarnya OFF menjadi berlogika benar (*true*) dan *power* dapat mengalir dengan kata lain instruksi hanya dapat berlogika benar jika terdapat kondisi OFF.



Gambar 2.11 Simbol kontak normally close (XIO).

* *Coil* (*output*), merupakan simbol keluaran pada pemrograman ladder logic. Keluaran dapat berupa motor, pompa, lampu dan relay. PLC menjalankan masukan pada lader dan mengaktifkannya atau mematikan keluaran tergantung pada kondisi dari masukannya. Apabila kontak pada ladder dalam keadaan true, maka power mengalir untuk mengoperasikan *coil*.



Gambar 2.12 Simbol coil.

* *Timer On Delay* (TON), Instruksi TON mengaktifkan keluaran setelah timer ON untuk interval preset time atau dengan kata lain timer menunda logika 1 pada output untuk sementara waktu. TON mulai menghitung ketika rung berlogika benar sampai beberapa kondisi ditentukan, yaitu nilai akumulasi sama dengan nilai preset atau kondisi rung menjadi salah. Status bit timer dapat digunakan pada ladder logic. Terdapat 3 macam status bit timer yaitu :
* *Enable* (EN)

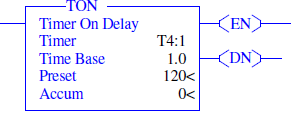
EN berlogika benar jika nilai *accumulated* sama dengan atau lebih besar dari nilai *preset*. Dan ketika instruksi *timer* berlogika salah maka *enable* juga bernilai salah.

* *Done*-bit (DN)

DN merubah *state* ketika nilai pada *accumulated* sama dengan nilai *preset* pada *timer*.

* *Timer Timing* (TT)

TT bernilai benar jika nilai *accumulated* berubah, yaitu jika *timer* sedang menghitung. Ketika *timer* tidak menghitung, nilai *accumulated* tidak berubah maka TT akan berlogika salah.



Gambar 2.13 Simbol timer on delay (TON).

* Instruksi *Timer Off Delay* (TOF) digunakan untuk menunda logika 0 pada output untuk sementara waktu. Perubahan *input trigger* dari logika 1 ke logika 0 akan mengaktifkan waktu tunda. Sesudah waktu tunda mencapai batasnya, *output* akan menghasilkan logika 0 sampai terjadi lagi perubahan logika pada masukkan *trigger*. Status bit *timer* dapat digunakan pada *ladder logic*. Terdapat tiga macam status bit pada *timer*, yaitu *Enable* (EN), *Timer-Timing* (TT) dan *Done*-bit (DN).
* *Enable* (EN)

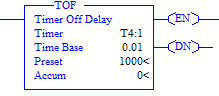
Bit *Enable* berlogika benar jika nilai *accumulated* sama dengan atau lebih besar dari nilai *preset*. Dan ketika instruksi *timer* berlogika salah maka bit *enable* juga bernilai salah.

* *Done*-bit (DN)

Bit *Done* merubah *state* ketika nilai pada *accumulated* sama dengan nilai *preset* pada *timer*.

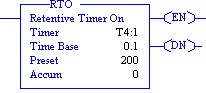
* *Timer Timing* (TT)

Bit *Timer-Timing* berlogika benar jika nilai *accumulated* berubah, yaitu jika *timer* sedang menghitung. Ketika *timer* tidak menghitung, nilai *accumulated* tidak berubah maka TT akan berlogika salah.



Gambar 2.14 Simbol timer off delay (TOF).

* Instruksi Retentive Timer On (TON) digunakan untuk mengaktifkan keluaran setelah timer ON untuk interval preset time atau dengan kata lain timer menunda logika 1 pada output untuk sementara waktu. TON mulai menghitung kerika rung berlogika benar, apabila nilai output berlogika 1 maka nilai output yang dicapai oleh timer disimpan, bahkan ketika kontinuitas logic atau daya hilang.



Gambar 2.15 Simbol retentive timer on (RTO).

* *Count Up* (CTU) adalah sebuah instruksi output yang fungsinya untuk menghitung naik nilai *accumulated* pada transisi *false to true* diinstruksi tersebut. Perhitungan internal akan menghitung setiap perubahan input dari kondisi 0 ke 1. *Output* akan aktif jika counter sudah mencapai batas perhitungan.

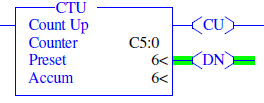
Terdapat beberapa status bit untuk fungsi CTU, yaitu :

* Count Up Enable (CU)

CU enable bit berlogika benar jika count up counter berlogika benar.

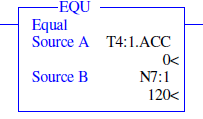
* Done-bit (DN)

DN berlogika benar ketika nilai accumulated sama dengan atau lebih besar dari nilai preset dari counter.

****

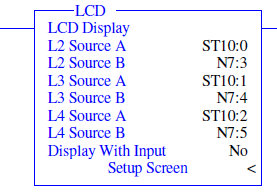
Gambar 2.16 Simbol count up.

* Instruksi Equal (EQU),Instruksi masukan ini membandingkan dua nilai spesifik. Jika kedua nilai tersebut sama, maka rung akan berlogika benar sehingga output juga bernilai benar (aktif).



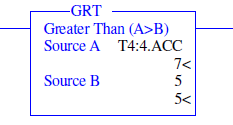
Gambar 2.17 Simbol equal (EQU).

* *Intruksi* LCD digunakan untuk menampilkan string atau nomor dan mendapatkan nilai dari keypad.



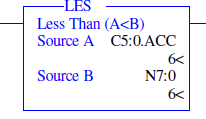
Gambar 2.18 Simbol LCD.

* *Greater Than* (GRT),digunakan untuk mengetes apakah nilai yang pertama lebih besar daripada nilah yang kedua.



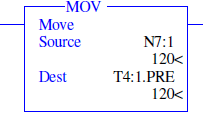
Gambar 2.19 Simbol greater than (GRT).

* *Less Than* (LESS),digunakan untuk mengetes apakah nilah yang pertama lebih kecil daripada nilai yang kedua.



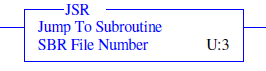
Gambar 2.20 Simbol less than (LESS).

* *Instruksi* MOV digunakan untuk memindah data dari sumber menuju tujuannya. Selama *rung* bernilai benar, instruksi pemindahan data akan meneliti.



Gambar 2.21 Simbol move (MOV).

* *Jump To Subroutine* (JSR)*,*Instruksi JSR menyebabkan kontrol untuk memulai mengeksekusi subroutine file secara terpisah dengan ladder diagram. JSR memindahkan program untuk dieksekusi kemudian didesain subroutine (SBB file number). Setelah mengeksekusi SBR kontrol proses akan mengikuti perintah yang ada pada JSR.



Gambar 2.22 Simbol jump to subroutine (JSR).

* *Block Transfer Read* (BTR) dan *Block Transfer Write* (BTW), Instruksi *block transfer* memungkinkan untuk mentransfer data (*word*) dari atau ke *block transfer module*. Ketika rung berlogika *true*, instruksi BTW akan memberi instruksi ke prosesor untuk menulis data yang disimpan di data file untuk dispesifikasikan alamat dari rack/group/modulenya. Sedangkan instruksi BTR memberi instruksi ke prosesor untuk membaca data dari alamat *rack/group/module* dan menyimpannya ke data file. Untuk memprogram instruksi BTR atau BTW harus disediakan sebuah prosesor dengan informasi yang terdapat pada kontrol *block* sebagai berikut :
* *Rack* adalah nomor rak I/O yang ditempati target modul.
* *Group* adalah nomor I/O yang menunjukkan posisi target modul I/O di *chasis* I/O.
* *Slot* adalah tempat nomor (0-1) di dalam group. Slot 0 adalah *Low* sedangkan *slot* 1 adalah *slot High*.
* *Module* adalah nomor slot dari group.
* *Control block.*
* *Data file.* 
  + 1. **PLC Omron CP1E-E20DRA**

*Programmable Logic Controller* (PLC) pada dasarnya adalah sebuah komputer yang khusus dirancang untuk mengontrol suatu proses atau mesin. Proses yang dikontrol ini dapat berupa regulasi variabel secara kontinyu seperti pada sistem-sistem servo atau hanya melibatkan kontrol dua keadaan (On/Off) saja tapi dilakukan secara berulang-ulang seperti umum kita jumpai pada mesin pengeboran, sistem konveyor, dan lain sebagainya.

Keterangan PLC Omron CP1E-E20DRA:

CP1E = Jenis PLC

E = Tipe unit (model dasar)

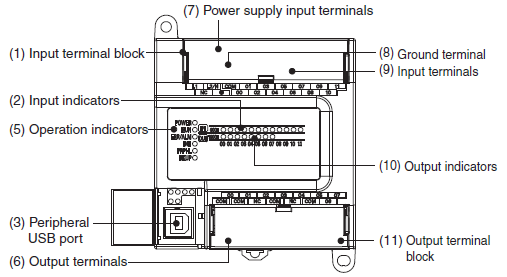
20 = Kapasitas Input/Output (20 I/O = 12 Input, 8 Output)

D = Mempunyai tegangan input DC.

R = Tipe outputnya adalah relay.

A = Input Power supply (catu daya) AC 100-240 volt.

Nama Bagian dan Fungsinya dari PLC CP1E-E20DR-A



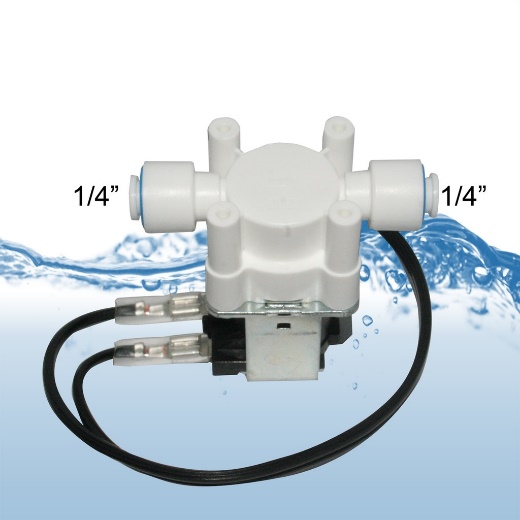
Gambar 2.23 Bagian PLC CP1E-E20SDRA.

Tabel 2.1 Fungsi dari masing-masing bagian pada PLC CP1E.

|  |  |  |
| --- | --- | --- |
| NO. | Nama | Fungsi |
| (1) | Input terminal block | Ini adalah blok terminal untuk input seperti masukan catu daya dan input push button, dll. |
| (2) | Input indicators (green) | Menampilkan status dari input. Sebuah indikator akan ON ketika input dalam kondisi ON. |
| (3) | Peripheral USB port | Sebagai penghubung ke komputer untuk pemrograman dan pemantauan oleh CX-Programmer untuk CP1E. |
| (5) | Operation indikators | Untuk mengetahui status operasi. |
| (6) | Output terminals | Penghubung output seperti ke relay, lampu, atau solenoid. |
| (7) | Power Supply input terminal | Digunakan sebagai terminal power supply (catu daya). |
| (8) | Ground terminal | Untuk mencegah sengatan listrik, tanah untuk 100 Q atau kurang. |
| (9) | Input terminals | Penghubung input seperti saklar, sensor, dll. |
| (10) | Output indicators (green) | Menampilkan status output. Indikator akan ON jika outputnya ON. |
| (11) | Output terminal block | Ini adalah blok terminal untuk input seperti relay, lampu, dll. |

* + 1. ***Solenoid Valve***

Katup atau *valve*, adalah sebuah alat untuk mengatur aliran suatu fluida dengan menutup, membuka atau menghambat sebagian dari jalannya aliran. Contoh yang mudah adalah keran air.

****

Gambar 2.24 *Solenoid valve.*

Solenoid *valve* akan bekerja bila kumparan/*coil* mendapatkan tegangan arus listrik yang sesuai dengan tegangan kerja (kebanyakan tegangan kerja solenoid *valve* adalah 100/200VAC dan kebanyakan tegangan kerja pada tegangan DC adalah 12/24V DC). Sebuah pin akan tertarik karena gaya magnet yang dihasilkan dari kumparan selenoida tersebut. Dan saat pin tersebut ditarik naik maka fluida akan mengalir dari ruang C menuju kebagian D dengan cepat. Sehingga tekanan di ruang C turun dan tekanan fluida yang masuk mengangkat diafragma. Sehingga katup utama terbuka dan fluida mengalir langsung dari A ke F.

Spesifikasi teknis solenoid valve dengan voltase 220V:

Ukuran dran inlet = 1/4’’

Outlet = 1/4”

Water pressure = 0.2-0.8 Mpa

Nilai daya = 4.8 W

Off = Closed, On = Open

* + 1. **Pompa Air**

Pompa air adalah alat yang digunakan untuk memindahkan cairan (*fluida*) dari suatu tempat ketempat yang lain, melalui media pipa (saluran) dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung terus menerus. Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara bagian hisap (*suction*) dan bagian tekan (*discharge*). Perbedaan tekanan tersebut dihasilkan dari sebuah mekanisme misalkan putaran roda *impeler* yang membuat keadaan sisi hisap nyaris vakum. Perbedaan tekanan inilah yang mengisap cairan sehingga dapat berpindah dari suatu *reservoir* ketempat lain. Pada jaman modern ini, posisi pompa menduduki tempat yang sangat penting bagi kehidupan manusia. Untuk memenuhi kebutuhan air sehari-hari dalam sebuah rumah, biasanya membutuhkan minimal 1 buah pompa air untuk mendistribusi air dari sumber keseluruh titik air (sumber air dapat berupa sumur atau penampungan air di bawah). Dalam kondisi ini memindahkan air dari sumber air di bawah dengan pompa untuk disimpan pada penampungan air.



Gambar 2.25 Pompa air.

Spesifikasi teknis pompa air :

Konsumsi daya = 1,5 W

Tegangan = 6-12V DC

Arus terukur maksimal = 0.3A

Laju Aliran maksimal = 280 liter/jam

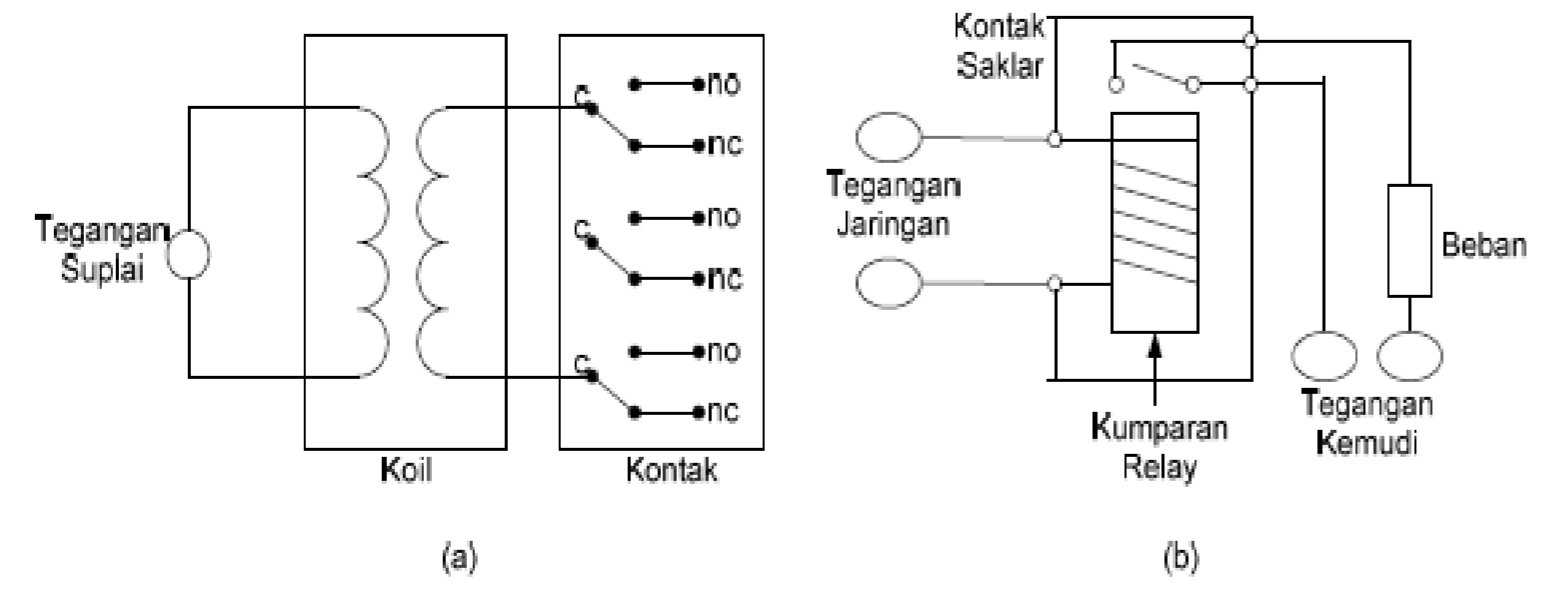
Ketinggian angkat = 2 m

* + 1. **Pengolah Sinyal Listrik**
* **Relay**

*Relay* merupakan suatu alat yang berfungsi sebagai *switch* elektronik dimana penggeraknya terbuat dari lilitan kawat tembaga. Pada dasarnya sebuah lilitan tembaga pada sebuah inti besi yang mana bila kedua ujungnya dihubungkan dengan sumber tegangan, maka akan timbul medan magnet pada inti besi tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 2.26.

Sedangkan kontak yang merupakan saklar terdapat dua macam kondisi dari kontak tersebut, yaitu :

* Normally Open (NO), yaitu kontak akan aktif pada saat koil di suplai tegangan.
* Normally Closed (NC), yaitu kontak akan aktif pada saat koil tidak di suplai tegangan.



Gambar 2.26 (a) Relay, (b) Bagian dalam relay.

Pada sebuah inti besi yang menimbulkan medan magnet akan menarik sebuah lempengan besi dari kontaktor, sehingga akan menyebabkan titik satu dengan titik lainnya akan tersambung. NC (mengunci).

Relay adalah komponen untuk penyambung saluran dan pengontrol sinyal, yang kebutuhan energinya relatif kecil. Relay ini biasanya difungsikan dengan elektromagnet yang dihasilkan dari kumparan. Pada awalnya relay ini digunakan pada peralatan telekomunikasi yang berfungsi sebagai penguat sinyal. Tapi sekarang sudah umum didapatkan pada perangkat kontrol, baik pada permesinan ataupun yang lainnya.

Cara kerja relay: Apabila pada lilitan dialiri arus listrik maka arus listrik tadi akan mengalir melalui lilitan kawat dan akan timbul medan magnet yang mengakibatkan pelat yang ada di dekat kumparan akan tertarik ataupun terdorong sehingga saluran dapat tersambung ataupun terputus. Hal ini tergantung apakah sambungannya NO atau NC. Bila tidak ada arus listrik maka pelat tadi akan kembali keposisi semula karena ditarik dengan pegas.

Simbol Relay:

Relay Normally Open

Relay Normally Closed

Kombinasi NO & NC

Penunjukkan angka pada relay mempunyai arti sebagai berikut:

Angka yang pertama menunjukkan contactor yang keberapa sedangkan angka yang kedua selalu bernomor ¾ untuk relay NO dan ½ untuk relay yang NC.

* + 1. **Catu Daya**

Catu daya merupakan suatu rangkaian yang paling penting dari sistem elektronika. Ada dua sumber catu daya yaitu sumber AC (alternating current) dansumber DC (direct current). Sumber AC yaitu sumber tegangan bolak-balik, sedangkan sumber tegangan DC merupakan sumber tegangan searah. Perangakat elektronik mestinya dicatu oleh suplay arus searah DC yang stabil agar berjalan dengan baik. Baterai atau accu adalah sumber catu daya DC yang paling banyak. Namun untuk aplikasi yang membutuhkan catudaya yang lebih besar, sumber beterai tidak cukup. Sumber catu daya yang besar adalah sumber bolak-balik AC dari PLN. Untuk itu diperlukan suatu perangkat catu daya yang dapat mengubah arus AC menjadi DC.