

BAHAN AJAR

PENGENDALI PLC



Disusun Oleh:

Drs. Wirawan Sumbodo,MT

**JURUSAN TEKNIK MESIN
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG 2008**

1. Aplikasi teknik Pengendalian dalam otomasi Industri

Teknologi Otomasi mulai ada sejak berabad-abad yang lalu, terutama sejak ditemukannya komponen *cam* dan *governor*. Pada tahun 1932, *Nyquist* mengembangkan suatu prosedur yang relatif sederhana untuk menentukan kestabilan sistem *loop* tertutup pada *basis respon loop* terbuka terhadap masukan tunak (*Steady State*) *Sinusoida*. Pada tahun 1934, *Hazien* memperkenalkan istilah servo mekanisme untuk sistem kontrol posisi, membahas desain servo mekanisme *relay* yang mampu mengikuti dengan baik masukan yang berubah. Pada dekade 1940-1950 pemakaian sistem kontrol otomatis telah berkembang, mulai tahun 1960 dengan berkembangnya perangkat peralatan (*plant*) dengan multi masukan dan multi keluaran maka sistem kontrol menjadi semakin kompleks.



Gambar 1, Penggunaan robot dalam sistem otomasi Industri

Selanjutnya secara berangsur angsur mulai memanfaatkan komponen elektronik-mekanik seperti *relay*, dan komponen elektronik seperti transistor. Perkembangan selanjutnya telah semakin cepat setelah ditemukannya komponen mikroelektronik dalam bentuk IC (*Integrated Circuit*) pada awal tahun 1960-an. Teknologi Otomasi semakin berkembang dengan pesat sejak munculnya mikroprosesor pada tahun 1973, sejak itu teknologi otomasi telah memasuki berbagai sektor kegiatan manusia, baik yang secara khusus misalnya di dalam dunia manufaktur, maupun secara umum dalam berbagai bentuk barang yang ada di sekeliling kita seperti Telefak, Mesin cuci dan sebagainya. Mesin cuci modern biasanya menggunakan sistem otomasi *loop* tertutup, sehingga proses pencuciannya dapat diprogram seperti yang diharapkan.



Gambar 2, Penggunaan robot dalam sistem otomasi Industri mobil

Teknologi Otomasi yang pada awalnya banyak diartikan sebagai pemakaian suatu sistem pengatur yang mampu menggerakkan suatu konstruksi mekanik (manipulator) secara mandiri tanpa campur tangan manusia, dewasa ini makin berkembang dengan dimasukkannya pengertian tentang kemampuan untuk mengatur pengolahan data secara mandiri. Dalam aplikasinya kegiatan proses produksi kedua cakupan pengertian di atas pada dasarnya sangat banyak digunakan. Pengertian teknologi otomasi yang didefinisikan sebagai penggunaan sistem pengatur yang mampu menggerakkan suatu manipulator atau konstruksi mekanik secara mandiri tanpa campur tangan manusia melahirkan suatu disiplin ilmu baru yang disebut sebagai mekatronika.

Proses produksi industri manufaktur mobil maupun sepeda motor di Indonesia sudah semakin pesat. Meski dengan jumlah karyawan yang sedikit namun mampu menghasilkan produk yang banyak dan dengan kualitas yang sama baiknya. Pada dasarnya teknologi otomasi dibedakan menjadi dua, yaitu *fixed automation* (otomasi tetap) dan *flexible Automation* (otomasi fleksibel). Konstruksi *fixed automation* biasanya masih menggunakan peralatan mekanik. Sedangkan *fleksibel automation* sudah menggunakan sistem pengatur berbasis komputer. Sistem pengatur berbasis komputer dirancang agar mudah dirubah sesuai dengan kebutuhan. Sebagai contoh penggunaan

robot industri, gerakan robot dapat dirubah sesuai dengan kebutuhan, juga penggunaan mesin perkakas CNC. Teknologi modern ditandai dengan penggunaan *fleksible automation* yang semakin meluas. *Fleksible Automation* akan terus berkembang sejalan dengan perkembangan mikroelektronika yang mendasar

Pemanfaatan teknologi otomasi pada proses produksi meliputi bidang yang sangat luas, dari kegiatan seperti pada bagian *Product Design, Production Planning dan Control, Inventory Control, Sales dan Marketing, Engineering, Industrial Engineering* banyak yang lebih berupa pengolahan secara otomatis data elektronik, sedangkan teknologi otomasi yang banyak di terapkan adalah dalam bidang produksi.

Pemanfaatan teknologi otomasi dalam proses produksi merupakan sebagian kecil saja dari penggunaan teknologi tersebut. Sebagian besar aplikasinya dimanfaatkan secara luas dalam kehidupan sehari-hari di masyarakat. Proses otomasi yang dapat kita lihat sehari-hari antara lain: mesin cuci otomatis, sistem pengisian tandon otomatis, pengering tangan otomatis, dan sebagainya. Dalam pembahasan selanjutnya, akan dibahas lebih jauh teknologi otomasi yang diterapkan dalam industri pengolahan serta pemesinan logam. Pembahasan akan lebih diarahkan pada teknologi otomasi dalam bentuk pengaturan gerak manipulator atau konstruksi mekanik yang terdapat dalam berbagai bentuk peralatan pabrik. Pengaturan yang akan dikembangkan berbasis pada sistem kontrol pneumatik, hidrolik, elektrik, dan juga mekanik.

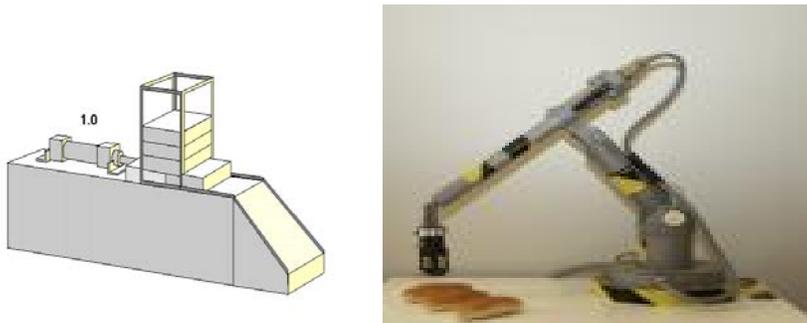
Rancangan konstruksi dari berbagai peralatan di atas ada yang dapat dibuat secara umum sehingga dapat diproduksi secara massal, seperti mesin perkakas CNC robot industri, berbagai jenis conveyor, AGV dsb nya, namun ada pula yang harus dirancang secara khusus untuk jenis pemakaian tertentu seperti mesin-mesin khusus, *Jig* dan *Fixtures*. Pemilihan peralatan yang sesuai dengan proses produksi yang hendak dilakukan merupakan tahap awal yang sangat menentukan tinggi rendahnya efisiensi proses produk tersebut. Pemilihan yang salah merupakan cacat bawaan yang akan sukar untuk diperbaiki nantinya, tanpa melakukan penggantian peralatan yang salah tersebut secara keseluruhan.

2. Otomasi Teknik Produksi

Setiap perusahaan selalu berusaha untuk efisien dan efektif dalam melakukan proses produksinya. Hal ini sesuai dengan prinsip ekonomi, yang bertujuan mendapatkan keuntungan yang sebesar-besarnya dengan biaya yang serendah-rendahnya. Salah satu upaya yang dilakukan perusahaan antara lain dengan cara

mengurangi biaya produksi, termasuk biaya tenaga kerja. Meningkatnya kualitas hidup berdampak pada gaji tenaga kerja terampil yang semakin mahal. Tenaga kerja terampil umumnya menuntut gaji yang besar. Padahal tenaga kerja terampil sebagai manusia pada umumnya memiliki keterbatasan seperti kelelahan, sakit, jenuh, bahkan kadang menuntut kenaikan gaji melalui demonstrasi yang dapat menghentikan aktivitas perusahaan. Dewasa ini perusahaan selalu berupaya untuk mengganti pekerjaan yang selama ini dilakukan oleh manusia untuk digantikan dengan mesin-mesin dalam rangka efisiensi dan peningkatan kualitas produksinya. Dengan kata lain banyak perusahaan melakukan otomasi produksinya.

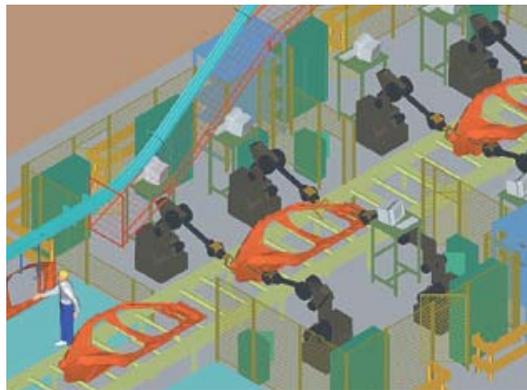
Istilah otomasi berasal dari otomatisasi, belakangan ini istilah otomatisasi tidak lagi banyak digunakan. Menurut *Thomas Krist* yang dikutip *Dines Ginting* (1993), ``Otomasi`` adalah mengubah penggerakan atau pelayanan dengan tangan menjadi pelayanan otomatis pada penggerakan dan gerakan tersebut berturut-turut dilaksanakan oleh tenaga asing (tanpa perantara tenaga manusia). Jadi otomasi menghemat tenaga manusia. Terutama suatu penempatan yang menguntungkan dari unsur-unsur pelayanan adalah mengurangi banyaknya gerakan-gerakan tangan sampai seminimum mungkin. Gerakan-gerakan yang biasa dilakukan manusia seperti menggeser, mengangkat, menempa, dan lain-lain telah dapat digantikan oleh gerakan aktuator mekanik, listrik, pneumatik, hidrolik, dan lain-lain. Masing-masing aktuator memiliki kelebihan dan kelemahan, misalnya lebih fleksibel dan bersih, namun mudah terbakar bila dibebani lebih. Pneumatik dapat dibebani lebih, bersih, dan aman, namun untuk menghasilkan udara bertekanan diperlukan peralatan mahal seperti kompresor dan katup-katup. Hidrolik mampu menghasilkan daya besar, namun memiliki keterbatasan temperatur dan cenderung kotor. Pemilihan aktuator tersebut akan selalu menyesuaikan dengan kebutuhan.



Gambar 3, Penggeser pneumatik dan robot industri yang siap menggantikan tenaga manusia

Penggantian tenaga manusia menjadi tenaga mesin akan meningkatkan produktivitas dan efisiensi kerja. Penggantian ini sangat tepat terutama pada industri bahan dasar, industri kimia dan tungku pengecoran logam bertemperatur tinggi, dimana akan mengurangi resiko kecelakaan kerja dan meningkatkan kenyamanan produksi. Faktor ini juga sangat menentukan kedayagunaan dan manfaat ekonomis dari produksi.

Pengalihan gerakan dari tenaga manusia ke mesin dapat dilakukan sebagian maupun keseluruhan. Otomasi sebagian berarti sistem masih memerlukan tenaga kerja untuk mengoperasikan mesin, sedangkan otomasi lengkap berarti semuanya dapat dikerjakan oleh mesin, tenaga manusia hanya bertindak sebagai programmer dari mesin tersebut. Dalam beberapa tahun ini perkembangan full otomasi telah berkembang pesat terutama pada industri manufaktur mobil maupun industri yang lain.

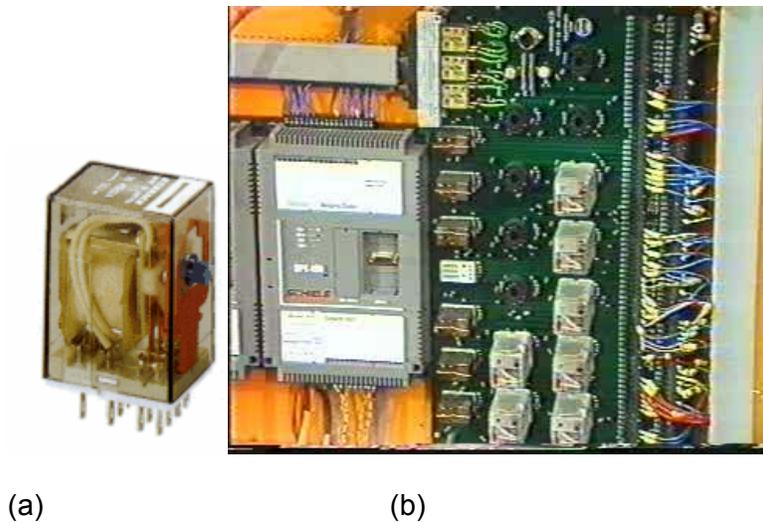


Gambar 4, Robot Industri dalam proses *manufacturing*

3. PLC (*Programmable Logic Controllers*)

3.1 Sejarah PLC

Secara historis PLC (*Programmable Logic Controllers*) pertama kali dirancang oleh Perusahaan General Motor (GM) sekitar pada tahun tahun 1968. PLC awalnya merupakan sebuah kumpulan dari banyak *relay* yang pada proses sekuensial dirasakan tidak fleksibel dan berbiaya tinggi dalam proses otomatisasi dalam suatu industri. Pada saat itu PLC penggunaannya masih terbatas pada fungsi-fungsi kontrol relay saja. Namun dalam perkembangannya PLC merupakan sistem yang dapat dikendalikan secara terprogram. Selanjutnya hasil rancangan PLC mulai berbasis pada bentuk komponen *solid state* yang memiliki fleksibilitas tinggi. Kerja tersebut dilakukan karena adanya prosesor pada PLC yang memproses program sistem yang diinginkan.



Gambar 5. Relay tunggal (a) dan Sistem *relay* pada mesin CNC (b)

Saat ini PLC telah mengalami perkembangan yang luar biasa, baik dari segi ukuran, kepadatan komponen serta dari segi fungsinya seiring perkembangan teknologi *solid state*. Beberapa perkembangan perangkat keras maupun perangkat lunak PLC antara lain: (a) Ukuran semakin kecil dan kompak, (b) Jenis instruksi/fungsi semakin banyak dan lengkap, (c) Memiliki kemampuan komunikasi dan sistem dokumentasi yang semakin baik, (d) Jumlah *input/output* yang semakin banyak dan padat, (f) Waktu eksekusi program yang semakin cepat, (g) Pemrograman relatif semakin mudah. Hal ini terkait dengan perangkat lunak pemrograman yang semakin *user friendly*, (h) Beberapa jenis dan tipe PLC dilengkapi dengan modul-modul untuk tujuan kontrol kontinu, misalnya modul ADC/DAC, PID, modul *Fuzzy* dan lain-lain.

Perusahaan PLC saat ini sudah memulai memproduksi PLC dengan beberapa ukuran, seperti jumlah *input/output*, instruksi dan kemampuan lainnya yang beragam. Perkembangan dewasa ini pada dasarnya dilakukan agar memenuhi dan memberikan solusi bagi kebutuhan pasar yang sangat luas. Sehingga mampu untuk menjawab permasalahan kebutuhan kontrol yang kompleks dengan jumlah *input/output* mencapai ribuan.

3.2 Pengenalan Dasar PLC

Pada dasarnya PLC (*Programmable Logic Controllers*) merupakan sistem relay yang dikendalikan secara terprogram. Kerja tersebut dilakukan karena adanya prosesor pada PLC yang memproses program yang diinginkan. PLC dilengkapi dengan *port* masukan (*inputport*) dan keluaran (*outputport*). Adanya masukan dan keluaran PLC

secara modul akan lebih mempermudah proses pengawatan (*wiring*) sistem. Pada dasarnya PLC terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Adapun jenis *hardware* dapat berupa unit PLC berbagai merek, seperti OMRON, Siemens, LG, dan lain lain, seperti contoh berikut berikut:



(a)



(b)

Gambar 6. PLC Omron type ZEN (a) dan Siemens (b)

Agar lebih mengenal fungsi dan cara kerja PLC pada umumnya, biasanya dibuat PLC *Training Unit* untuk keperluan pelatihan bagi siswa maupun praktisi industri agar lebih mendalami dan memahaminya



Gambar 7. PLC Training Unit

3.3. Instruksi-instruksi dasar PLC

Instruksi (perintah program) merupakan perintah agar PLC dapat bekerja seperti yang diharapkan. Pada setiap akhir program harus di instruksikan kalimat END yang oleh PLC dianggap sebagai batas akhir dari program. Instruksi END tidak ditampilkan pada tombol operasional *programming console*, akan tetapi berupa sebuah fungsi yaitu FUN (01).

3.3.1 LOAD (LD) dan LOAD NOT (LD NOT)

LOAD adalah sambungan langsung dari line dengan logika pensakelarnya seperti sakelar NO, sedangkan LOAD NOT logika pensakelarnya seperti sakelar NC. Instruksi ini dibutuhkan jika urutan kerja pada sistem kendali hanya membutuhkan satu kondisi logic saja untuk satu *output*. Simbol *ladder diagram* dari LD dan LD NOT seperti Gambar 8 di bawah ini:

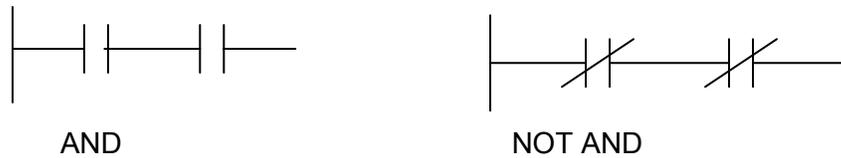


Gambar 8. Simbol logika LOAD dan LOAD NOT.

3.3.2 AND dan NOT AND (NAND)

Jika memasukkan logika AND maka harus ada rangkaian yang berada di depannya, karena penyambungannya seri. Logika pensaklaran AND seperti sakelar NO dan NOT AND seperti saklar NC. Instruksi tersebut dibutuhkan jika urutan kerja sistem kendali

lebih dari satu kondisi logic yang terpenuhi semuanya untuk memperoleh satu *output*. Simbol *ladder diagram* dari AND dan NOT AND seperti Gambar 9. di bawah ini:



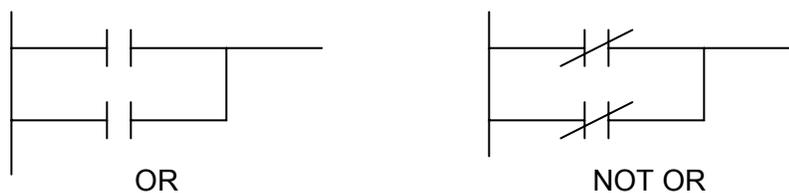
Gambar 9. Simbol logika AND dan NOT AND

Tabel 1. Logika AND dan NOT AND (NAND)

S_1	S_2	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

3.3.3 OR dan NOT OR

OR dan NOT OR (NOR) dimasukkan seperti saklar posisinya paralel dengan rangkaian sebelumnya. instruksi tersebut dibutuhkan jika urutan kerja sistem kendali membutuhkan salah satu saja dari beberapa kondisi logic terpasang paralel untuk mengeluarkan satu *output*. Logika pensaklaran OR seperti saklar NO dan logika pensaklaran NOT OR seperti saklar NC. Simbol *ladder diagram* dari OR dan OR NOT seperti gambar 11. di bawah ini:

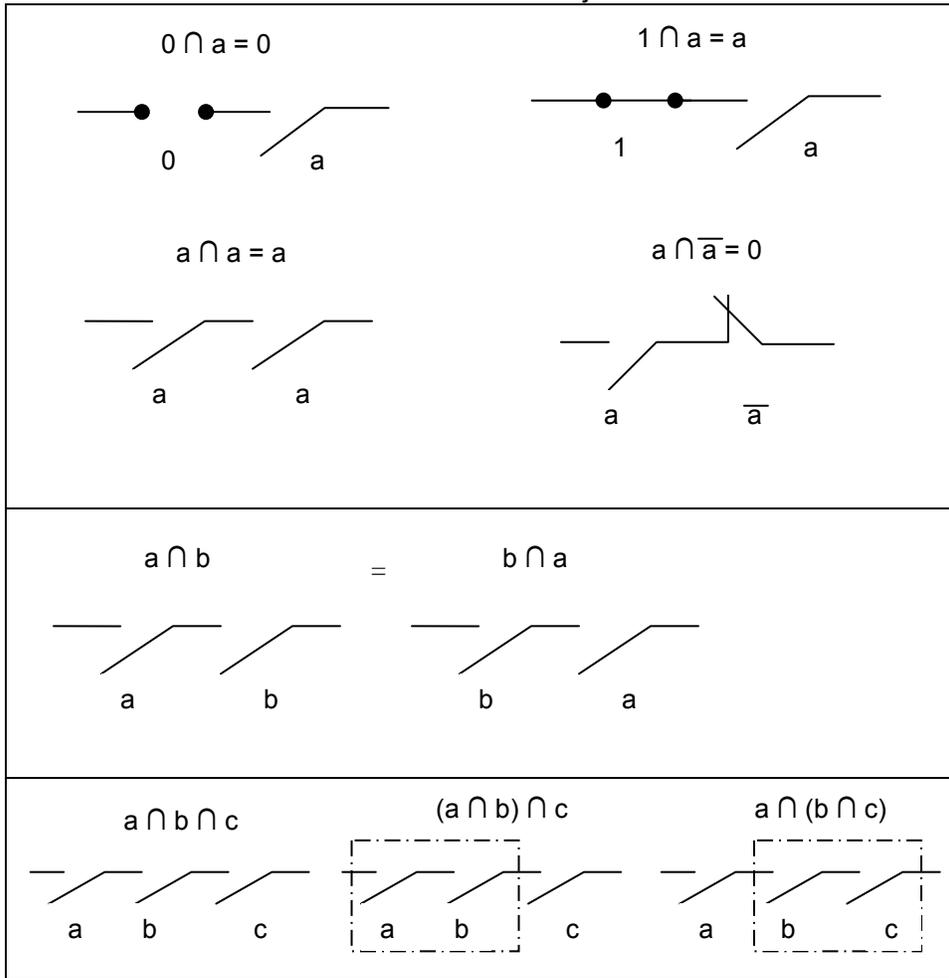


Gambar 10. Simbol logika OR dan NOT OR

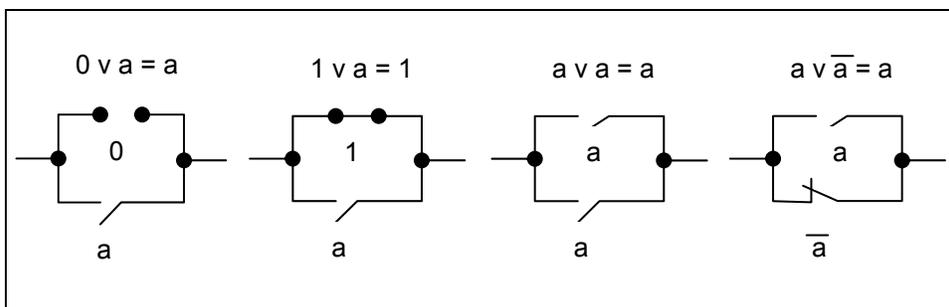
Tabel 2. Logika OR dan NOT OR (NOR)

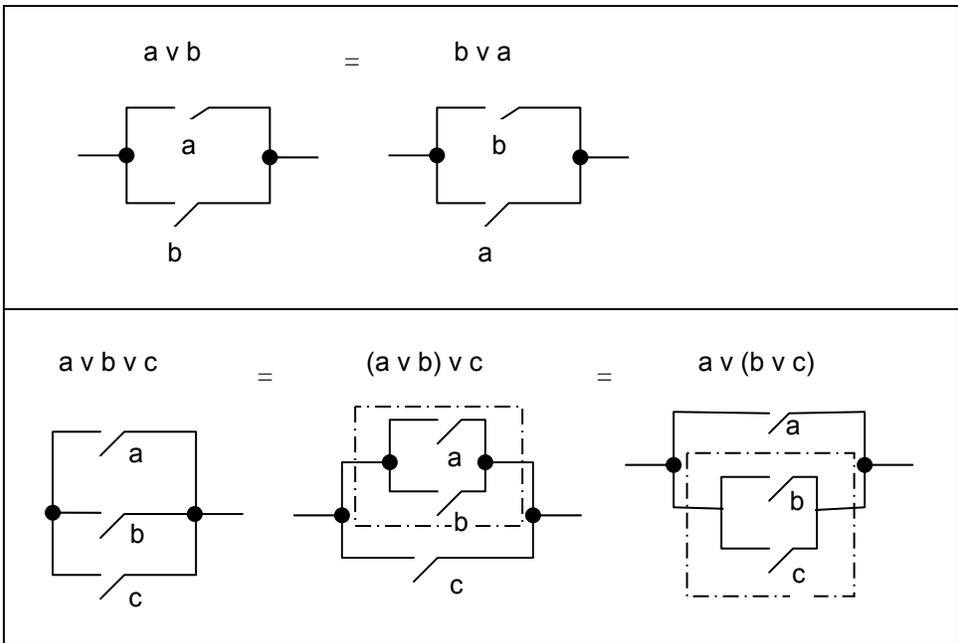
S_1	S_2	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Tabel 3. Aturan aljabar saklar AND



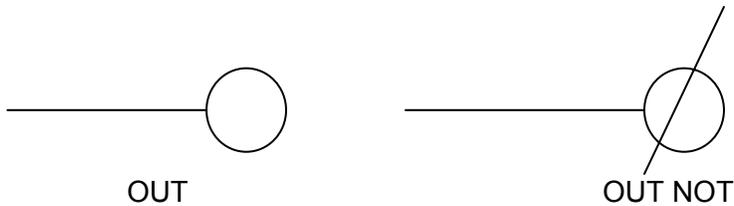
Tabel 4. Aturan aljabar saklar OR





3.3.4 OUT dan OUT NOT

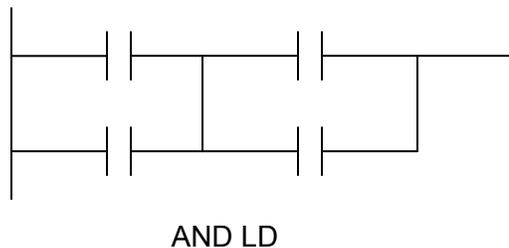
Digunakan untuk mengeluarkan *Output* jika semua kondisi logika ladder diagram sudah terpenuhi. Logika pensaklaran OUT seperti sakelar NO dan logika pensaklaran OUT NOT seperti saklar NC. Adapun Simbol *ladder diagram* dari OUT dan OUT NOT



Gambar 11. Simbol logika OUT dan OUT NOT.

3.3.5 AND LOAD (AND LD)

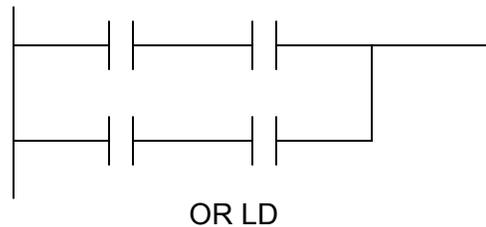
Digunakan untuk kondisi logika ladder diagram yang khusus dimaksudkan untuk mengeluarkan satu keluaran tertentu. Simbol ladder diagram dari AND LD seperti gambar 12. di bawah ini:



Gambar 12. Simbol logika AND LOAD.

3.3.6 OR LOAD (OR LD)

Digunakan untuk kondisi logika *ladder diagram* yang khusus dimaksudkan untuk mengeluarkan satu keluaran tertentu. Simbol ladder diagram dari OR LD seperti gambar 13 di bawah ini:

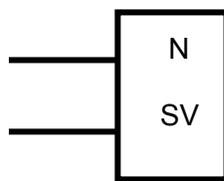


Gambar 13. Simbol logika OR LOAD.

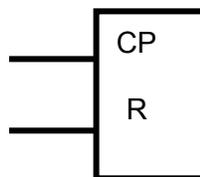
3.3.7 TIMER (TIM) dan COUNTER (CNT)

- Jumlahnya bergantung dari masing-masing tipe PLC. Jika suatu nomor sudah dipergunakan sebagai TIMER/COUNTER, maka nomor tersebut tidak boleh lagi dipakai lagi sebagai TIMER/COUNTER yang lain.
- Nilai TIMER/COUNTER bersifat menghitung mundur dari nilai awal yang ditetapkan oleh program. Setelah hitungan tersebut mencapai angka nol, maka kontak NO TIMER/COUNTER akan bekerja.
- TIMER mempunyai batas hitungan antara 0000 sampai 9999 dalam bentuk BCD (*binary Code Decimal*) dan dalam orde sampai 100 ms. Sedangkan COUNTER mempunyai orde angka BCD dan mempunyai batas antara 0000 sampai dengan 9999.

Simbol ladder diagramnya seperti Gambar 14 di bawah ini:



TIMER
Keterangan :
N : Nomor T/C
SV : Set Value

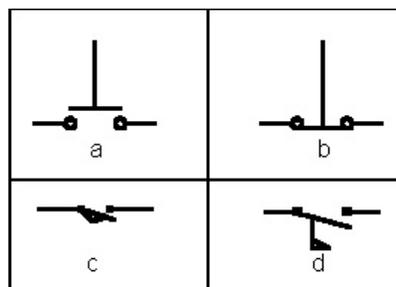
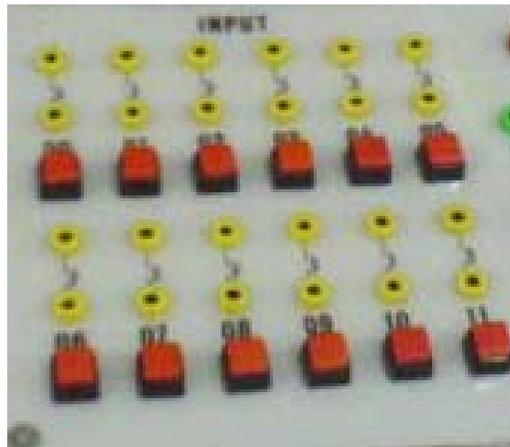


COUNTER
Keterangan :
CP : Pulsa
R : Reset

Gambar 14. Simbol logika TIMER dan COUNTER.

3.4. Device Masukan

Device masukan merupakan perangkat keras yang digunakan untuk memberikan sinyal kepada modul masukan. Sistem PLC memiliki jumlah *device* masukan sesuai dengan sistem yang diinginkan. Fungsi dari *device* masukan untuk memberikan perintah khusus sesuai dengan kinerja *device* masukan yang digunakan, misalnya untuk menjalankan atau menghentikan motor. Dalam hal tersebut seperti misalnya *device* masukan yang digunakan adalah *push button* yang bekerja secara *Normally Open (NO)* ataupun *Normally Close (NC)*. Ada bermacam-macam *device* masukan yang dapat digunakan dalam pembentukan suatu sistem kendali seperti misalnya: *selector switch*, *foot switch*, *flow switch*, *level switch*, *proximity sensors* dan lain-lain. Gambar15. memperlihatkan macam-macam simbol masukan.



Keterangan :

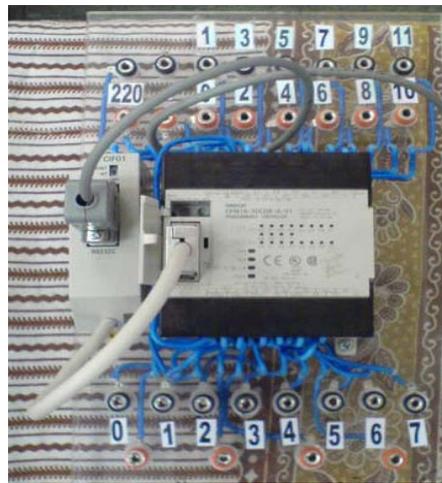
- a. NO *Pushbutton*
- b. NC *Pushbutton*

- c. NO *Limit Switch*
- d. NO *Flow Switch*

Gambar 15. Contoh simbol *device* masukan

3.5. Modul Masukan

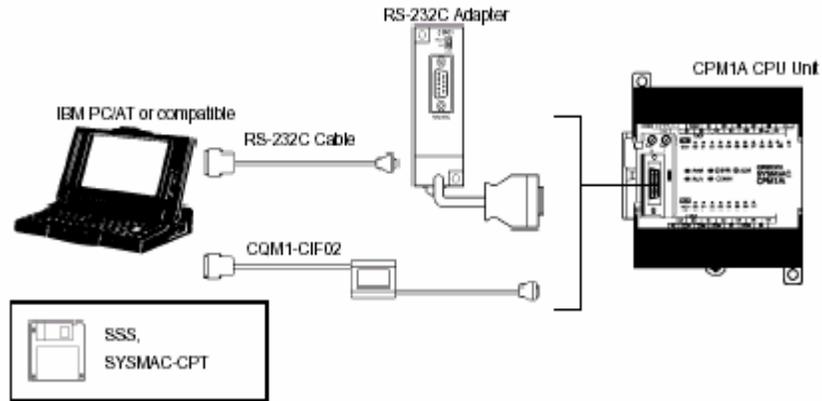
Modul masukan adalah bagian dari sistem PLC yang berfungsi memproses sinyal dari *device* masukan kemudian memberikan sinyal tersebut ke prosesor. Sistem PLC dapat memiliki beberapa modul masukan. Masing-masing modul mempunyai jumlah terminal tertentu, yang berarti modul tersebut dapat melayani beberapa *device* masukan. Pada umumnya modul masukan ditempatkan pada sebuah rak. Pada jenis PLC tertentu terdapat modul masukan yang ditempatkan langsung satu unit dengan prosesor ataupun catu daya dan tidak ditempatkan dengan sistem rak. Gambar 16 memperlihatkan modul masukan atau keluaran yang penempatannya menggunakan rak.



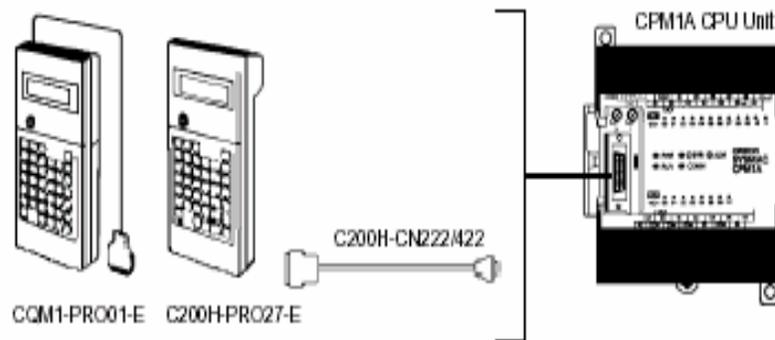
Gambar 16. Slot Modul masukan atau keluaran PLC

3.6. Device Masukan Program

Device masukan program berfungsi sebagai sarana untuk memasukkan atau mengisi program ke dalam prosesor PLC yang disebut dengan pengisi program (*program loader*). *Program Loader* sering disebut sebagai *device programmer* yaitu alat yang digunakan untuk melakukan pengisian program ke CPU. *Device programmer* membuat program PLC menjadi lebih *fleksibel*. *Device programmer* memperbolehkan pemakai untuk melakukan perubahan program kendali baru (modifikasi) atau memeriksa benar atau tidaknya program yang telah diisikan ke dalam memori. Hal ini sangat membantu untuk keperluan perawatan ketika timbul masalah terhadap sistem. Jenis-jenis *device programmer* yang sering digunakan adalah *desktop*, *handheld programmer* dan *device programmer* yang memang khusus dibuat oleh pembuat PLC. Gambar 17 dan Gambar 18. memperlihatkan contoh gambar *device programmer*.



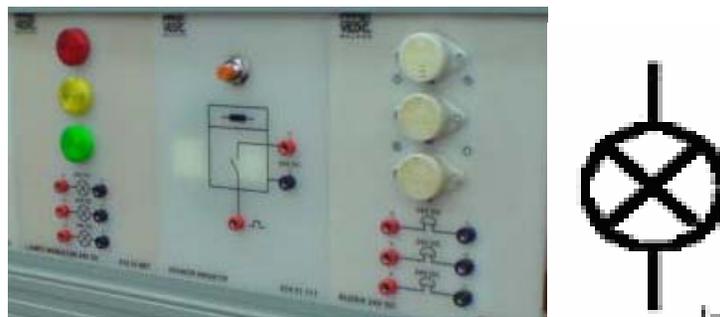
Gambar 17. Desktop.



Gambar 18. Handled Programmer. (OMRON)

3.7. Device Keluaran

Device keluaran adalah komponen-komponen yang memerlukan sinyal untuk mengaktifkan komponen tersebut. Sistem PLC mempunyai beberapa *device* keluaran seperti motor listrik, lampu indikator, sirine. Gambar 19. memperlihatkan contoh simbol dari *device* keluaran yang sering digunakan.



(a) Simbol lampu indikator



(b) Motor listrik dan simbolnya



(c) Katup penggerak Solenoid

Gambar 19. Contoh *device* keluaran dan simbolnya

3.8. Modul Keluaran

PLC dapat mempunyai beberapa modul keluaran tergantung dari ukuran dan aplikasi sistem kendali. *Device* keluaran disambungkan ke modul keluaran dan akan aktif pada saat sinyal diterima oleh modul keluaran dari prosesor sesuai dengan program sistem kendali yang telah diisikan ke memorinya. Catu daya yang digunakan untuk mengaktifkan *device* keluaran tidak langsung dari modul keluaran tetapi berasal dari catu daya dari luar, sehingga modul keluaran sebagai sakelar yang menyalurkan catu daya dari catu daya luar ke *device* keluaran.

3.9. Perangkat Lunak PLC

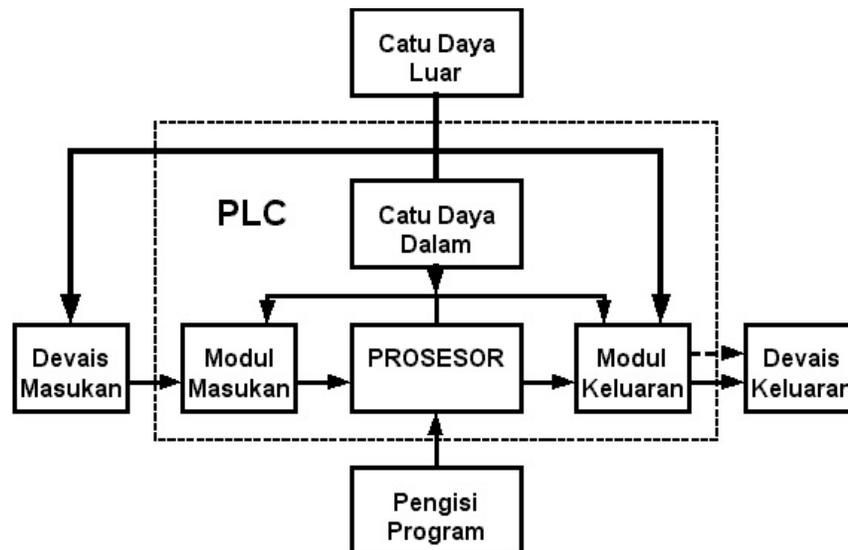
Pemrograman PLC terdiri dari instruksi-instruksi dasar PLC yang berbentuk logika pengendalian sistem kendali yang diinginkan. Bahasa programmeran biasanya telah disesuaikan dengan ketentuan dari pembuat PLC itu sendiri. Dalam hal ini setiap pembuat PLC memberikan aturan-aturan tertentu yang sudah disesuaikan dengan programmeran CPU yang digunakan PLC.

3.10. Perangkat Keras PLC

Sistem PLC menggunakan prinsip pemodulan yang memiliki beberapa keuntungan, seperti komponen-komponennya dapat ditambah, dikurangi ataupun dirancang ulang untuk mendapatkan sistem yang lebih *fleksibel*.

Sistem PLC memiliki tiga komponen utama yaitu unit prosesor, bagian masukan/keluaran, dan *device pemrograman*. Diagram kerja tiga komponen utama di atas, akan dijelaskan lebih rinci dengan gambar diagram blok sistem PLC seperti terdapat pada Gambar 20.

Urutan kerja dari gambar diagram blok di atas dimulai dari *device* masukan yang akan memberikan sinyal pada modul masukan. Sinyal tersebut diteruskan ke prosesor dan akan diolah sesuai dengan program yang dibuat. Sinyal dari prosesor kemudian diberikan ke modul keluaran untuk mengaktifkan *device* keluaran.



Gambar 20. Diagram Blok PLC

3.11. Ladder Logic

Ladder logic adalah bahasa programmeran dengan bahasa grafik atau bahasa yang digambar secara grafik. Diagram ini menyerupai diagram dasar yang digunakan logika kendali sistem kontrol panel dimana ketentuan instruksi terdiri dari koil-koil, NO, NC dan dalam bentuk penyimbolan. Programmeran tersebut akan memudahkan programmer dalam mentransisikan logika pengendalian khususnya bagi programmer yang memahami logika pengendalian sistem kontrol panel. Simbol-simbol tersebut tidak

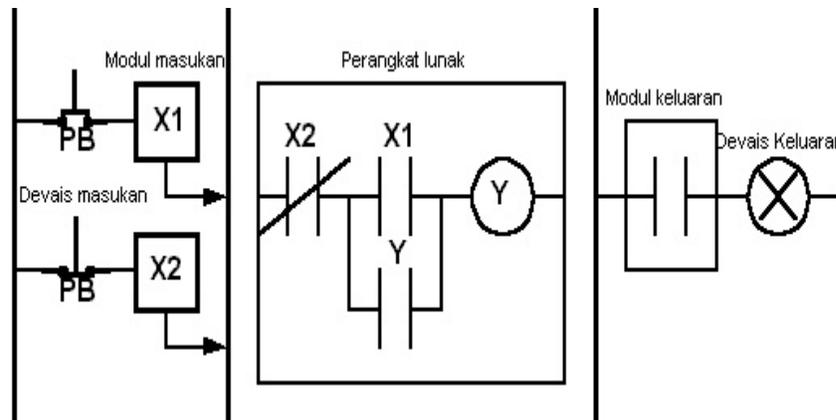
dapat dipresentasikan sebagai komponen, tetapi dalam programmerannya simbol-simbol tersebut dipresentasikan sebagai fungsi komponen sebenarnya.

3.12. Hubungan *Input/Output (I/O)* dengan Perangkat Lunak

Pada saat pemrogram (*programmer*) bekerja dengan bahasa *ladder logic*, *programmer* harus mengerti hubungan I/O dengan perangkat lunak. Untuk memudahkan pemahamannya, titik masukan modul masukan dapat dianggap sebagai *koil relay* yang masing-masing memiliki alamat tertentu. *koil relay* masukan berada di luar perangkat sehingga tidak dapat tergambar di perangkat lunak dan hanya memiliki kontak-kontak pada perangkat lunak. Banyaknya titik-titik keluaran terletak di modul keluaran. Untuk lebih mempermudah pemahaman mengenai hubungan I/O dengan perangkat lunak Gambar 21 memperlihatkan gambar hubungan antara I/O dengan perangkat lunak.

Gambar 21 memperlihatkan bahwa apabila *push button* 1 ditekan maka unit input X1 menjadi ON. Sesuai dengan prinsip pemahaman bahwa titik masukan sebagai *koil relay* yang mempunyai kontak di perangkat lunak, sehingga jika keadaan ON maka sinyal mengalir menuju modul masukan (dengan anggapan pemahaman bahwa terdapat koil) hal tersebut mengakibatkan kontak dari unit input di dalam perangkat lunak akan bekerja. Peristiwa itu tersebut mengakibatkan koil keluaran perangkat lunak menerima sinyal tersebut sehingga unit *output* sebagai kontak koil akan bekerja.

Apabila lampu indikator sebagai *device* keluaran, kejadian tersebut mengakibatkan lampu menyala. Karena sebagai *device* masukan berupa *push button* 1 ON saat ditekan saja (NO) maka untuk membuat lampu itu menyala terus, koil keluaran perangkat lunak memiliki *internal relay* yang dapat digunakan sebagai pengunci (*holding*). Sinyal selanjutnya mengalir melalui *holding relay* tersebut dan lampu akan menyala terus dan akan mati apabila *pushbutton* 2 ditekan karena terputusnya tegangan dalam hal ini karena *pushbutton* 2 sebagai NC.



Gambar 21. Hubungan antara I/O dengan perangkat lunak.

3.13. Processor

Prosesor adalah bagian pemroses sistem PLC yang membuat keputusan logika. Keputusan yang telah dibuat berdasarkan program tersimpan dalam memori. Prosesor adalah bagian dari *Central Processing Unit* (CPU) dari PLC yang menerima, menganalisa, memproses dan memberikan informasi ke modul keluaran. Di dalam CPU PLC dapat dibayangkan seperti sekumpulan ribuan relay. Hal tersebut bukan berarti di dalamnya terdapat banyak relay dalam ukuran yang sangat kecil tetapi berisi rangkaian elektronika digital yang dapat sebagai kontak NO dan NC relay.

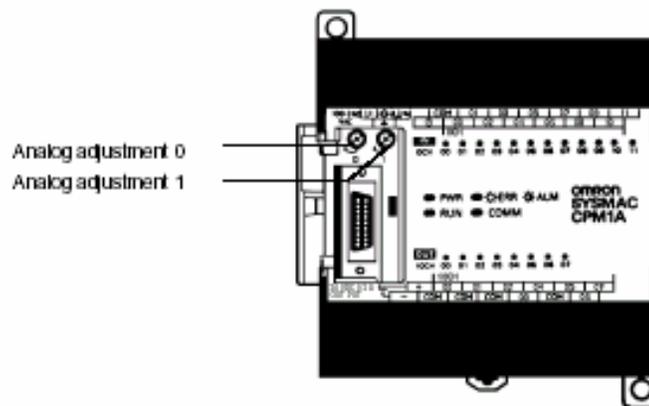
Memori berfungsi sebagai tempat dimana informasi tersebut disimpan. Ada bermacam-macam jenis serpih memori dalam bentuk *Integrated Circuit* (IC). Masing-masing jenis memori memiliki keuntungan dan kerugian dan dipilih untuk spesifikasi yang terbaik untuk aplikasinya.

Salah satu jenis memori yang digunakan dalam CPU PLC adalah *Random Access Memory* (RAM). Kerugian jenis memori tersebut adalah diperlukannya catu daya untuk menjaga agar memory tetap bekerja. Pada aplikasi PLC diperlukan catu daya cadangan yang digunakan untuk menjaga agar isi dari memori tidak hilang apabila tiba-tiba catu daya hilang. RAM digunakan untuk keperluan memori karena RAM mudah diubah dengan cepat ketika dibandingkan dengan jenis memori yang lain. RAM disebut juga sebagai memori baca/tulis, karena RAM dapat dibaca dan ditulis data untuk disimpan di RAM.

Read Only Memory (ROM) adalah jenis memori yang semi permanen dan tidak dapat diubah dengan pengubah program. Memori tersebut hanya digunakan untuk

membaca saja dan jenis memori tersebut tidak memerlukan catu daya cadangan karena isi memori tidak akan hilang meskipun catu daya terputus

Programmable Read Only Memory (PROM) adalah jenis lain dari memori yang bekerja hampir menyerupai ROM, dengan satu pengecualian yaitu bisa di program. PROM di rancang untuk diisi dengan program yang terprogram. Apabila data dapat diubah, maka dapat diadakan programmeran. Programmeran ulang dari PROM membutuhkan perlengkapan khusus yaitu *PROM Programmer* dimana PLC sendiri tidak dapat melakukannya. Gambar 22. memperlihatkan contoh CPU PLC yang menggunakan sistem RAM.



Gambar 22. CPU PLC (OMRON)

3.14 Data dan Memory PLC

3.14.1 Aturan dasar penulisan memori PLC adalah :

- *Word* atau *channel* yang terdiri dari 16 bit, ditulis **XXX**
- *Bit* atau *contact* yang terdiri dari 1 bit, ditulis **XXXXXX**, dua angka yang paling belakang menunjukkan nomor *contact* dan sisa angka yang depan menunjukkan nomer *channel*.

3.14.2 Memori PLC

3.14.2.1 *Internal Relay*

Internal relay (IR) mempunyai pembagian fungsi seperti IR *input*, IR *output* dan juga IR *work area* (untuk pengolahan data pada program). IR *input* dan IR *output* adalah IR yang berhubungan dengan terminal *input* dan *output* pada PLC. Sedangkan IR *work*

area tidak dihubungkan ke terminal PLC, akan tetapi berada dalam *internal memory* PLC dan fungsinya untuk pengolahan logika program.

Terdapat juga IR yang fungsinya untuk *SYSMAC BUS area*, *Special I/O Unit area*, *Optical I/O unit area*, dan *Group 2 High density I/O unit area*.

3.14.2.2 Special Relay

Special relay (SR) merupakan *relay* yang menghubungkan fungsi-fungsi khusus seperti *flag* (misalnya: instruksi penjumlahan terdapat kelebihan digit pada hasilnya [*carry flag*], kontrol bit PLC, informasi kondisi PLC, dan *system clock* (pulsa).

3.14.2.3 Auxiliary Relay (AR)

Auxiliary relay terdiri dari *flags* dan *bit* untuk tujuan khusus. Dapat menunjukkan kondisi PLC yang disebabkan oleh kegagalan sumber tegangan, kondisi *special I/O*, kondisi *input/output* unit, kondisi CPU PLC, memori PLC dan lain-lain.

3.14.2.4 Holding Relay

Holding relay (HR) dapat difungsikan untuk menyimpan data (*bit-bit* penting) karena tidak hilang walaupun sumber tegangan PLC mati.

3.14.2.5 Link Relay

Link relay (LR) digunakan untuk *data link* pada *PLC link system*. *Link system* digunakan untuk tukar-menukar informasi antar dua PLC atau lebih dalam satu sistem kendali yang saling berhubungan satu dengan yang lainnya dengan menggunakan PLC minimum dua unit.

3.14.2.6 Temporary Relay

Temporary relay (TR) berfungsi untuk menyimpan sementara kondisi logika program pada *ladder diagram* yang mempunyai titik percabangan khusus.

3.14.2.7 Timer/Counter

Timer/counter (T/C) untuk mendefinisikan suatu waktu tunda /*time delay* (*timer*) ataupun untuk menghitung (*counter*). Untuk timer mempunyai orde 100 ms, ada yang mempunyai orde 10 ms yaitu TIMH(15). Untuk TIM 000 sampai dengan TIM 015 dapat dioperasikan secara *interrupt* untuk mendapatkan waktu yang lebih presisi.

3.14.2.8 Data Memory

Data memory (DM) berfungsi untuk menyimpan data-data program karena isi DM tidak akan hilang (*reset*) walaupun sumber tegangan PLC mati. Macam-macam DM adalah sebagai berikut:

- > DM *read/write*
 - Pada DM *read/write* data-data program dapat dihapus dan ditulis oleh program yang dibuat, sehingga sangat berguna untuk manipulasi data program.
- > DM *special I/O unit*
 - DM *special I/O* berfungsi untuk menyimpan dan mengolah hasil dari *special I/O unit*, mengatur dan mendefinisikan sistem kerja *special I/O unit*.
- > DM *history Log*
 - Pada DM *history log* disimpan informasi-informasi penting pada saat PLC terjadi kegagalan sistem operasionalnya. Pesan-pesan kesalahan system PLC yang di simpan berupa kode-kode angka tertentu.
- > DM *link test area*
 - DM *link test area* berfungsi untuk menyimpan informasi-informasi yang menunjukkan status dari *system link* PLC.
- > DM *setup*

DM *setup* berfungsi untuk kondisi *default* (kondisi kerja saat PLC aktif). Pada DM inilah kemampuan kerja suatu PLC didefinisikan untuk pertama kalinya sebelum PLC tersebut diprogram dan dioperasikan pada suatu sistem kontrol. Tentu saja setup PLC tersebut disesuaikan dengan sistem kontrol yang bersangkutan.

3.14.2.9 Upper Memory

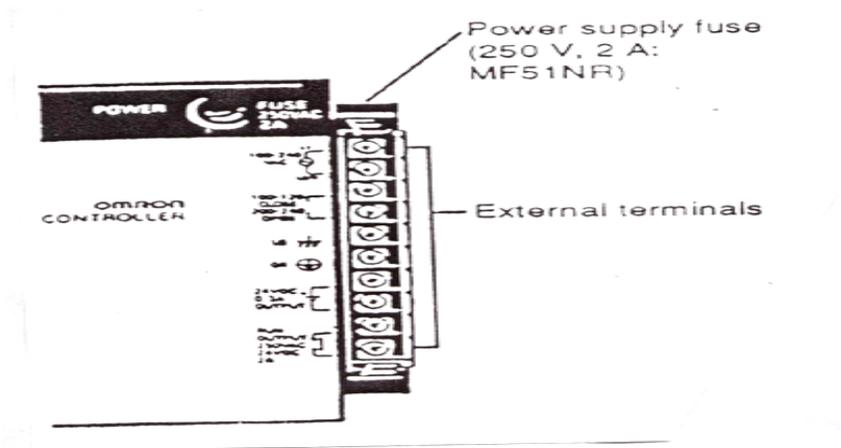
Upper memory (UM) berfungsi untuk menyimpan dan menjalankan program. Kapasitas tergantung dari pada masing-masing tipe PLC yang dipakai.

- ⇒ Semua memori (selain DM dan UM) dapat dibayangkan sebagai *relay* yang mempunyai koil, kontak NO dan NC. *Timer dan Counter* juga dapat dibayangkan seperti pada umumnya dan mempunyai kontak NO dan NC.
- ⇒ DM tidak mempunyai kontak, hanya ada *channel/word* saja. DM dapat difungsikan untuk menyimpan data-data penting yang tidak boleh hilang waktu sumber tegangan mati atau memanipulasi program.

- ⇒ Memori yang sifatnya dapat menyimpan data program jika listrik mati adalah DM dan HR, sedangkan memori yang lainnya akan hilang.
- ⇒ Programmeran PLC ada dua macam yaitu dengan diagram *ladder* dan bahasa *mnemonic*. Programmeran biasanya membuat diagram *ladder* terlebih dahulu dan kemudian baru menterjemahkannya dalam bahasa *mnemonic*, atau bisa juga langsung digambar ladder diagram pada layar monitor.

3.14.2.10 Catu Daya (*Power Supply*)

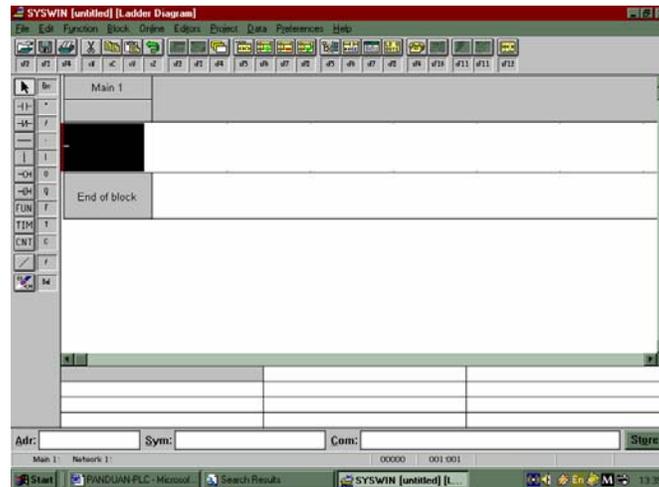
Sistem PLC memiliki dua macam catu daya dibedakan berdasarkan fungsi dan operasinya yaitu catu daya dalam dan catu daya luar. Catu daya dalam merupakan bagian dari unit PLC itu sendiri sedangkan catu daya luar yang memberikan catu daya kepada seluruh bagian dari sistem termasuk didalamnya untuk memberikan catu daya pada catu daya dalam dari PLC. Catu daya dalam mengaktifkan proses kerja PLC. Besarnya tegangan catu daya yang dipakai disesuaikan dengan karakteristik PLC. Bagian catu daya dalam PLC sama dengan bagian-bagian yang lain di mana terdapat langsung pada satu unit PLC atau terpisah dengan bagian yang lain dari atau sistem rak. Gambar catu daya yang sering digunakan dengan sistem rak diperlihatkan pada gambar 23.



Gambar 23. Catu Daya.

3.15. Programman PLC dasar Omron Dengan Komputer

Programman PLC dasar merk OMRON menggunakan bahasa program dari OMRON juga yaitu SYSWIN. Tampilan menu utama dari program SYSWIN dapat dilihat pada gambar berikut.



Gambar 24. Tampilan menu utama program SYSWIN (OMRON)

Beberapa perintah program yang penting dan perlu dipahami adalah sebagai berikut:

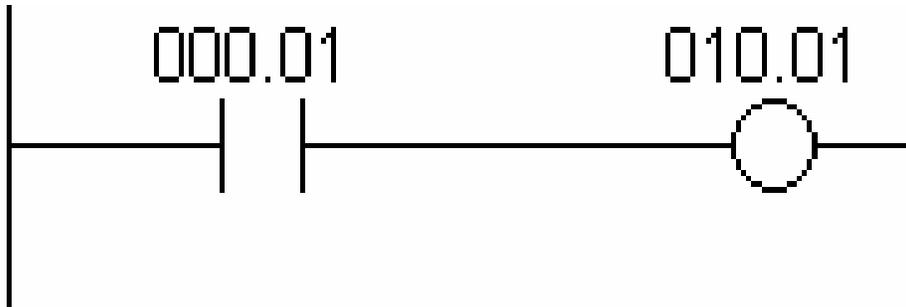
- > **Connect**
Connect merupakan perintah program untuk penyambungan antara komputer dengan PLC.
- > **Upload Program**
Merupakan perintah untuk melihat isi program dalam PLC
- > **Down Load Program**
Merupakan perintah untuk mentransfer program yang telah dibuat ke dalam PLC
- > **Run**
Perintah untuk menjalankan program yang telah di tranfer ke PLC
- > **Stop**
Perintah untuk menghentikan program yang sedang dijalankan di PLC
- > **Monitoring**
Perintah untuk melihat kondisi pada saat PLC bekerja

3.16 Cara pengoperasian SYSWIN

3.16.1 Pembuatan diagram ladder (diagram tangga)

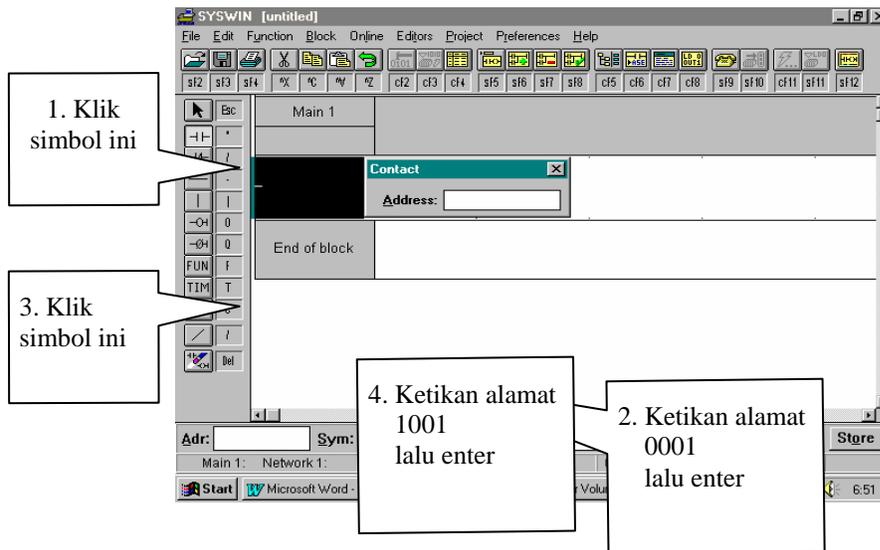
Pembuatan diagram ladder dapat dilakukan dengan cara klik kiri mouse pada menu perintah sesuai dengan yang dikehendaki kemudian memindahkan mouse ke layar tampilan yang dituju. Langkah selanjutnya memberikan alamat yang

dikehendaki pada perintah tersebut. Sebagai contoh membuat diagram ladder berikut:



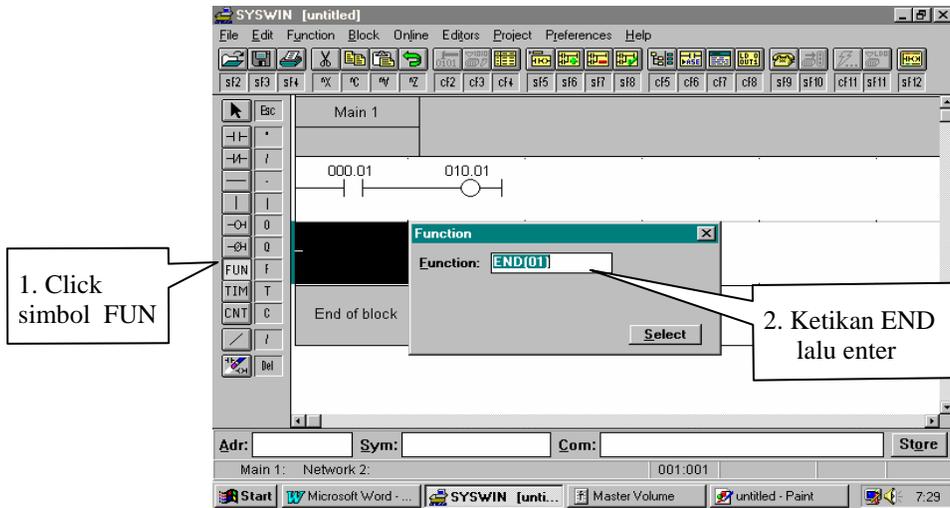
Gambar 25. Tampilan Ladder Diagram

Langkah sbb:



Gambar 26. Pembuatan diagram ladder

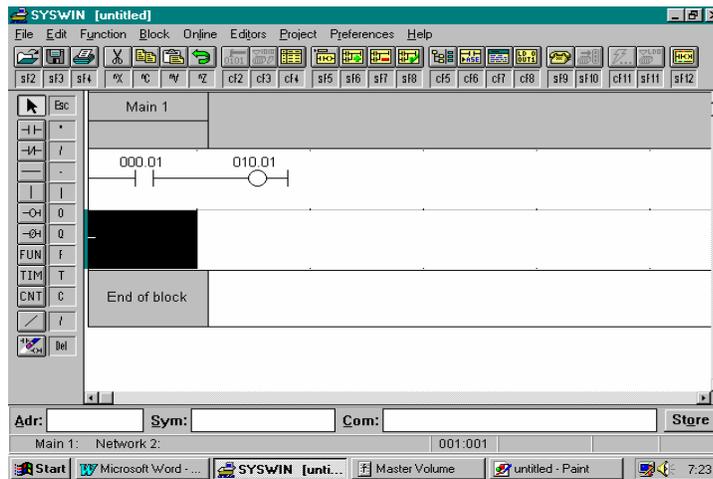
1. Untuk membuat ladder baru lagi di bawahnya maka posisikan mouse pad **End of blok** kemudian klik dua kali maka posisi **End of blok** akan turun dan kita dapat menggunakannya baris kosong tersebut untuk membuat diagram ladder baru.
2. Untuk mengakhiri program maka harus diakhiri dengan perintah **END** sebelum program tersebut dijalankan caranya sebagai berikut:



Gambar 27. Akhir dari diagram tangga menggunakan END

Setelah sebuah program diagram ladder dibuat kemudian untuk menjalankannya atau memasukkannya ke dalam PLC harus melewati langkah sebagai berikut:

1. Pastikan PLC sudah tersambung dan ter-conect dengan PLC
2. Sorot menu **Online**
3. Pilih perintah **Download Program** lalu enter
4. Pada menu **Online** pilih **Mode**

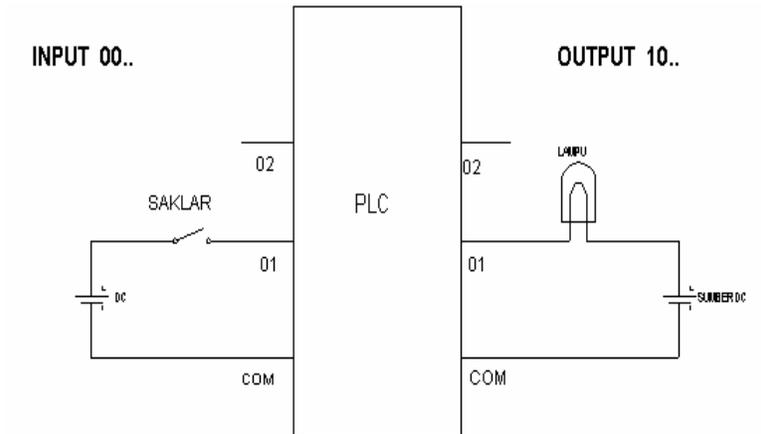


Gambar 28. Akhir dari diagram tangga menggunakan END

5. RUN untuk menjalankan program dalam PLC
6. STOP untuk menghentikan program

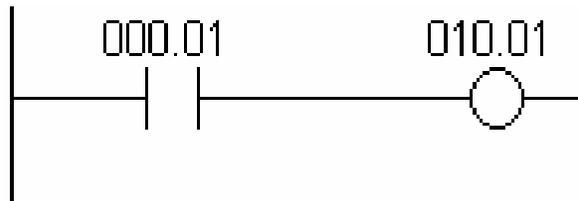
7. Untuk keperluan monitoring jalannya program dapat dipilih pada menu *Online* yaitu *Monitoring*

3.16.2 Cara Penyambungan Dan Logika Laddernya



Gambar 29. Penyambungan perangkat *Input*, *Output*, PLC, Catu daya

Pada gambar di atas apabila dibuat program dengan menggunakan diagram ladder sebagai berikut :

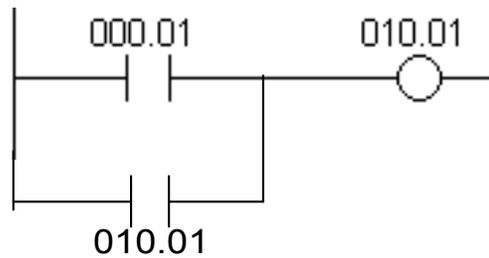


Gambar 30. Ladder diagram

Maka kerja dari rangkaian tersebut adalah:

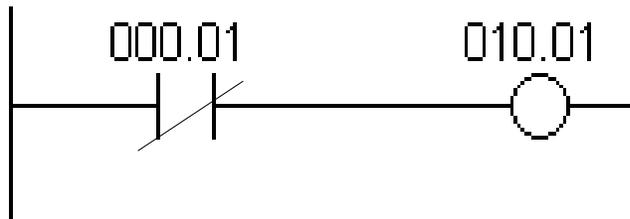
1. Jika input saklar **ditekan** maka output berupa lampu akan **menyala**
2. Tetapi jika sakelar **dilepas** maka lampu juga akan **mati**

Apabila dikehendaki lampu tetap menyala meskipun sakelar hanya sekali tekan maka perlu ditambahi dengan pengunci sebagai berikut:



Gambar 31. Ladder diagram dengan pengunci.

Kebalikan dari kerja rangkaian di atas (Gambar 31) apabila dibuat program dengan menggunakan diagram ladder sebagai berikut :



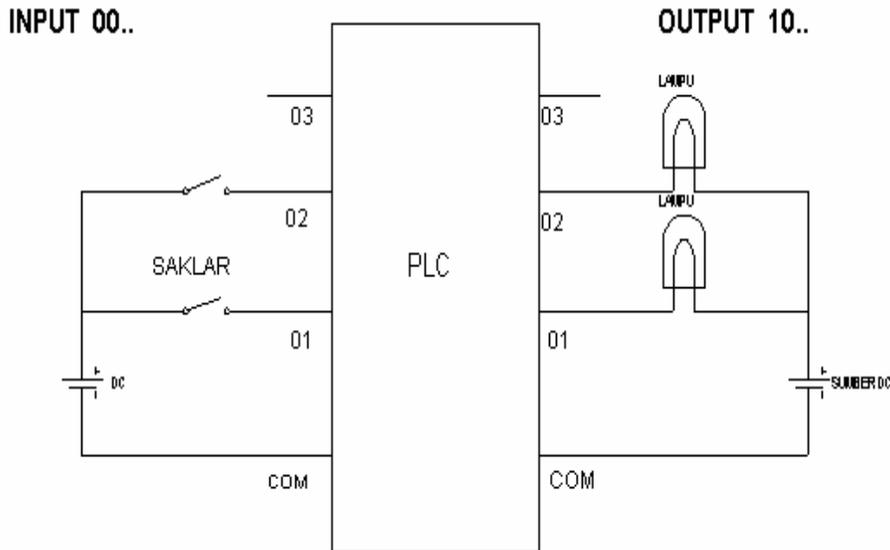
Gambar 32. Ladder diagram kebalikan dari kerja rangkaian di atas

Maka kerja dari rangkaian tersebut adalah:

Jika input saklar **tidak ditekan** maka output berupa lampu akan **menyala**

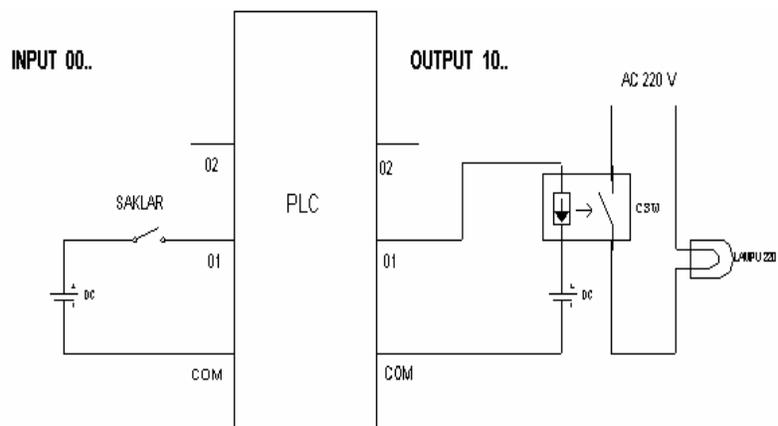
Jika input saklar **ditekan** maka output berupa lampu akan **mati**

Untuk penyambungan yang lebih dari satu *channel* maka cara penyambungan adalah sebagai berikut:



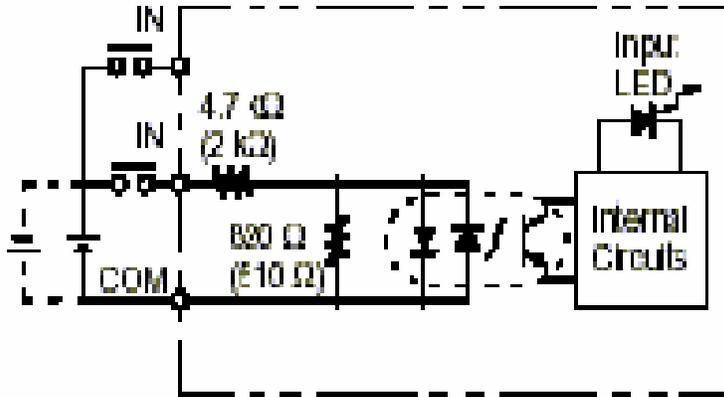
Gambar 33. Cara penyambungan perangkat *Input* dan *Output* lebih dari satu *channel*.

Oleh karena keterbatasan PLC dimana spesifikasi dari masukannya dan keluarannya adalah dengan tegangan dan arus yang kecil maka cara penyambungan dari peralatan keluarannya jika menggunakan lampu untuk tegangan dan arus tinggi adalah menggunakan peralatan *relay* seperti gambar di bawah ini. Untuk arus dan tegangan yang lebih besar dapat menggunakan *Magnetic Contactor*. Tegangan yang disambungkan ke *relay* ataupun *Magnetic Contactor* disesuaikan dengan tegangan dari *relay* atau *Magnetic Contactor* tersebut.



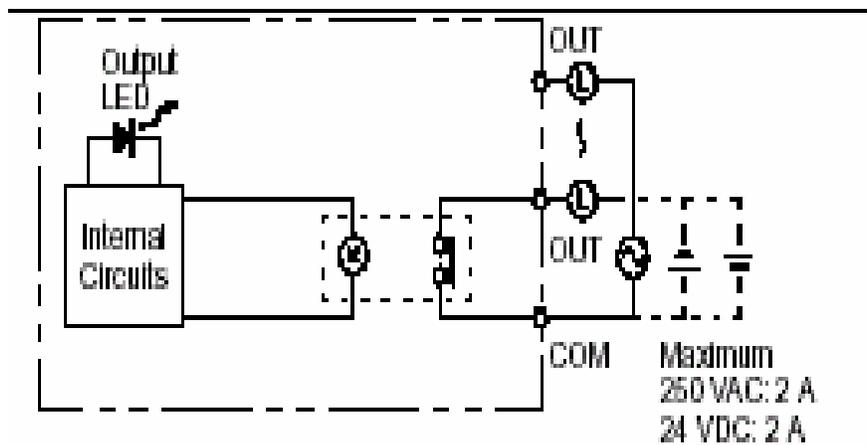
Gambar 34. Penambahan *relay* untuk memperbesar kemampuan arus.

Rangkaian *Input* dan *Output* di dalam Unit CPU PLC OMRON CPM1A-XXCDR dapat dilihat pada Gambar 35 dan Gambar 36 di bawah ini.



Note Figures in parentheses are for IN0000 to IN0002.

Gambar 35. Rangkaian Input Unit CPU PLC OMRON CPM1A-XXCDR

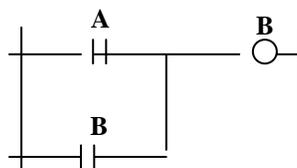


Gambar 36. Rangkaian Output dalam Unit CPU PLC OMRON CPM1A -XXCDR. (351352 modul pelatihan PLC OMRON)

3.17 Penggunaan Fungsi Bit Kontrol

3.17.1 LATCHING / SELF HOLDING

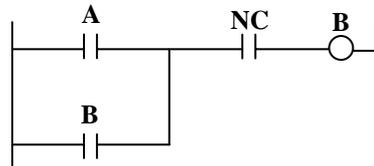
Fungsi ini berfungsi sebagai penahan dirinya sendiri pada suatu rangkaian ladder PLC.



Gambar 37. Rangkaian latching

Pada gambar diagram ladder di atas bisa kita lihat, antar *output* B baris pertama dengan input B pada baris kedua memiliki alamat yang sama yaitu B, yang berarti bahwa bila ada tegangan sekejap dari rangkaian di atas maka *output* akan tetap ON.

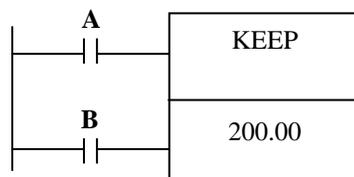
Sehingga cara mematakannya (OFF) yaitu dengan memberikan input berupa ladder NC (*normally close*), seperti pada gambar di bawah ini



Gambar 38. Diagram ladder latching

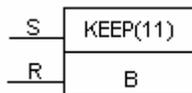
3.17.2 KEEP

Fungsi *keep* sama seperti pada fungsi *latching* hanya saja lebih sederhana dalam pembuatan ladernya karena telah memiliki tanda sendiri pada pemrogramannya seperti pada `SYSWIN` atau `SYSMAC`.



Gambar 39. Tanda KEEP

Secara umum fungsi *keep* memiliki aturan seperti gambar dibawah ini.

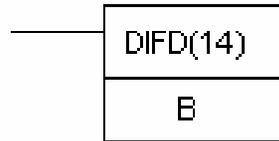
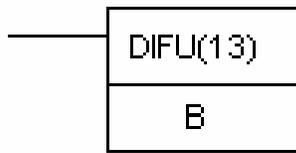


Gambar 40. Aturan KEEP

Dengan S sebagai set, yang berfungsi sebagai pemicu adanya KEEP. R sebagai reset yaitu untuk mengembalikan ke keadaan semula dan B atau *bit* yaitu memberikan *output* berupa *latching* atau mempertahankan dirinya sendiri.

3.17.3 DIFU dan DIFD

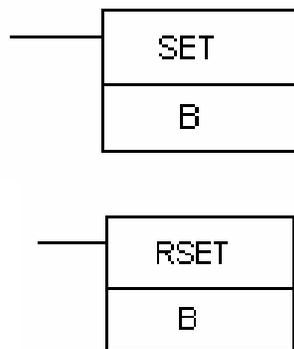
DIFU atau *Differentia Up* dan DIFD (*Differentia Down*) digunakan untuk menciptakan bit ON pada satu siklus. Perbedaannya terletak pada aktif tinggi (up) dan aktif rendah (down).



Gambar 41. DIFU dan DIFD

3.17.4 SET dan RESET

Penggunaan set dan reset hampir mirip dengan *latching* tetapi biasanya digunakan pada aplikasi yang panjang sehingga akan memudahkan dalam pembuatan program.



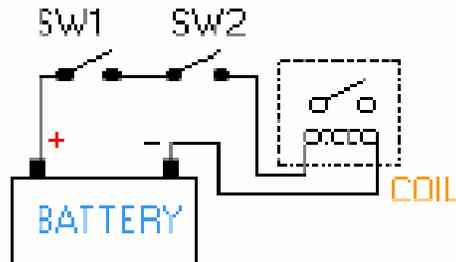
Gambar 42. SET dan RESET

Fungsi set memerintahkan agar *output* selalu dalam keadaan ON. Berbeda dengan Reset, memerintahkan agar *output* dalam keadaan OFF.

3.18 Contoh Aplikasi Dan Pembuatan Diagram Ladder Menggunakan Syswin

3.18.1 Aktivasi Koil

Sekarang akan kita bandingkan diagram ladder di atas dengan menggunakan aplikasi sirkuit di bawah ini.



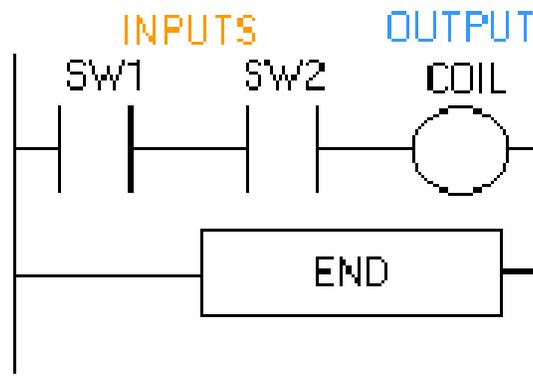
Gambar 43. Rangkaian switch relay

Keterangan :

SW1 = Switch 1

SW2 = Switch 2

Pada gambar sirkuit di atas, koil akan menjadi aktif bila terminal + (*positif*) dan terminal – (*negative*) pada baterai terhubung. Kita dapat mensimulasikan atau mentransfer ke dalam diagram ladder. Pada gambar di atas terdiri dari dua *input* dan satu *output*. Input terdiri dari dua *switch* yaitu *switch 1* dan *switch 2* sedangkan *output* adalah koil.

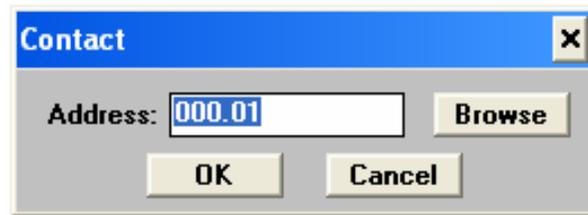


Gambar 44. Diagram ladder aplikasi koil

Pembuatan diagram ladder pada *software Syswin*

1. klik gambar *open contact* atau  pada sisi kiri yaitu *toolbar* simbol.

2. tempatkan pada work area, kemudian muncul *form dialog* untuk pengisian *addres /* alamat



isikan alamat untuk *switch 1* (missal : 000.01)

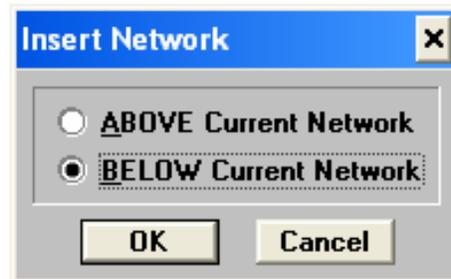
klik kembali  pada *toolbar symbol* dan masukan alamat / *addres* yang berbeda dengan *address switch 1* (missal 000.02)

Klik *symbol*  atau *open output* pada *toolbar symbol*
Tempatkan pada area kerja, sehingga akan muncul *dialog form*



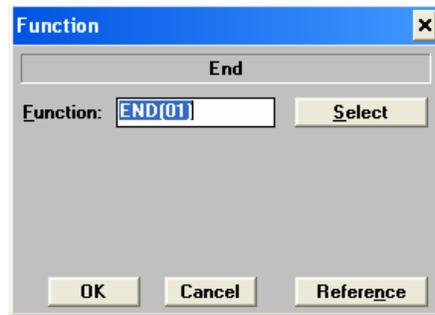
Isikan alamat *output*, missal 000.01, dan tekan OK

Klik Menu → *Block* → *Insert Block*, dan akan muncul *form dialog bok*

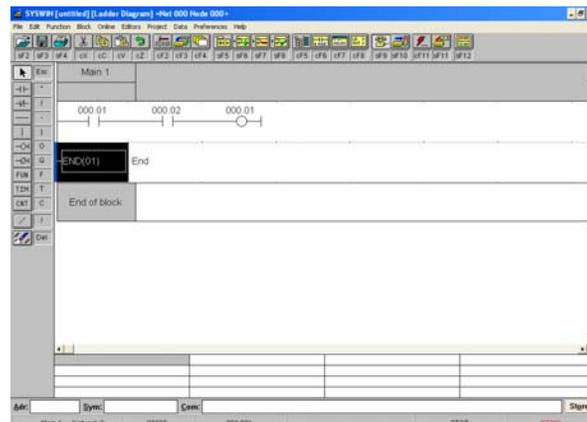


Pilih **BELOW Current Network**, Ok

Klik Menu → *Function* → *Basic instruction* → *Program Control Instruction*, akan muncul dialog.

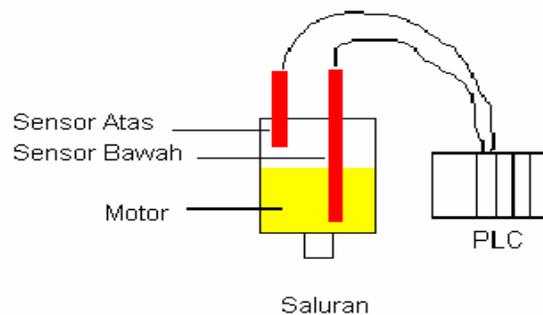


Pilih END(01) dan tekan Ok
 Hasil program seperti dibawah ini



3.18.2 Minyak Pelumas

Aplikasi di bawah ini adalah mengontrol minyak pelumas yang dikeluarkan dalam sebuah tangki. Dengan menggunakan dua sensor yaitu sensor bagian atas dan sensor bagian bawah. Masing masing sensor bertugas untuk mengontrol posisi ketinggian minyak pelumas.



Gambar 45. Aplikasi PLC

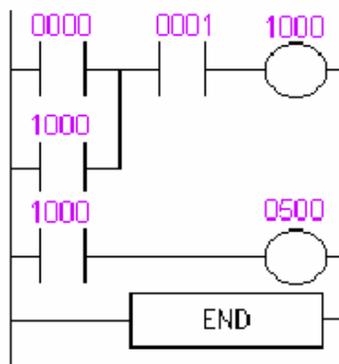
Pada saat volume minyak pelumas menyentuh sensor bawah motor akan memompa minyak pelumas menuju tangki dan berhenti ketika minyak pelumas menyentuh sensor bagian atas. Maka dibutuhkan 3 *input* dan *output*. 2 berupa *input*

dan 1 berupa *output*. Sensor dapat berupa sensor optik apabila tercelup dalam cairan atau minyak pelumas maka akan OFF disini motor akan OFF sehingga tidak memompa minyak. Apabila sensor tidak tercelup dalam minyak pelumas maka sensor akan ON dan menghidupkan motor dan memompa minyak pelumas ke dalam tangki. Sistem ini disebut dengan NC (*Normally closed*)

Tabel 5. Pengalamatan peralatan *input* dan *output*

Input	Adres	Output	Adres	Internal Relay
Sensor Bawah	000.00	Motor	010.00	010.00
Sensor Atas	000.01			

Setelah membagi alamat-alamat, *input* dan *output* maupun *internal relay*, kemudian kita membuatnya dalam diagram ladder.



Gambar 46. Diagram ladder

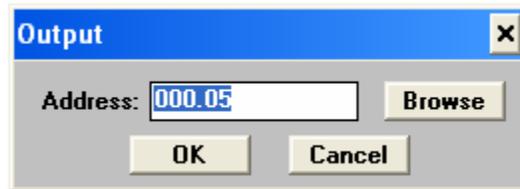
Pembuatan diagram ladder pada *Software Syswin 3.4* sebagai berikut

1. Klik gambar *open contact*  pada sisi kiri yaitu *toolbar* simbol.
2. Tempatkan pada *work area*, kemudian muncul *form dialog* bok untuk pengisian *adres* / alamat

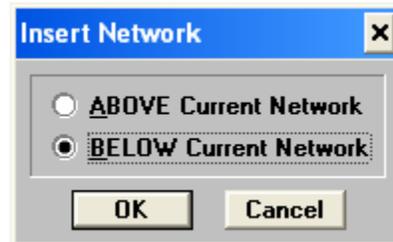


3. Isikan alamat untuk sensor atas (missal : 000.00)

4. Klik kembali  pada *toolbar symbol* dan Klik pada *work area* kemudian masukan alamat / adres yang berbeda dengan address sensor atas (missal 000.01)
5. Klik symbol  atau *open output* pada *toolbar symbol*
6. Tempatkan pada area kerja, sehingga akan muncul dialog *form dialog*.



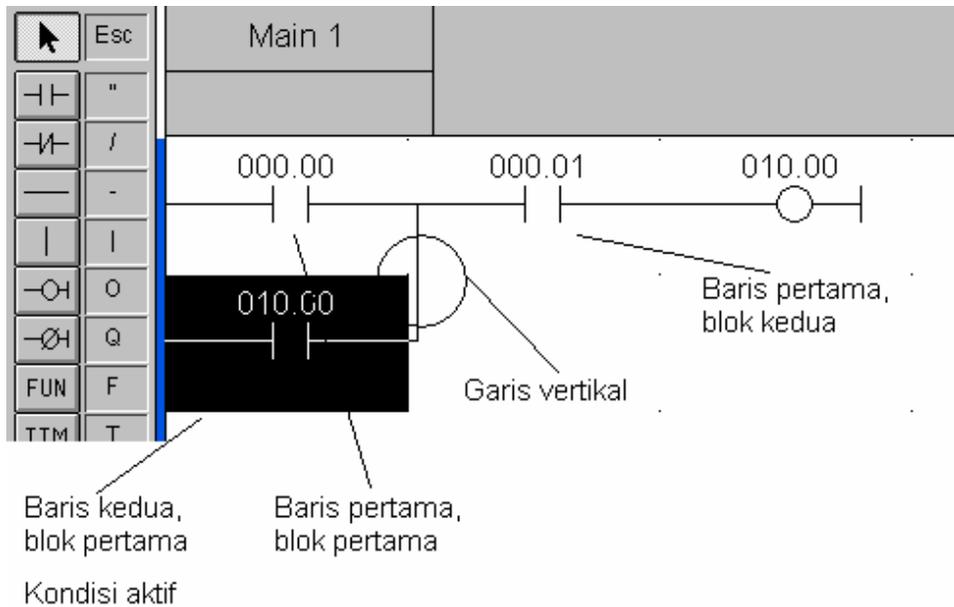
7. Isikan alamat *output*, missal 000.05, dan tekan OK (disini bukan sebagai *output* tetapi *internal relay*)
8. Klik Menu → *Block* → *Insert Block*, dan akan muncul *form dialog bok*



9. Pilih **BELOW *Current Network***, Ok
10. Klik gambar *open contact*  pada sisi kiri yaitu *toolbar* simbol.
11. Tempatkan pada *work area*, kemudian muncul *form dialog bok* untuk pengisian *adres / alamat*



12. Isikan alamat yang sama dengna *internal relay* diatas untuk sensor atas yaitu 000.05
13. Klik pada *selection tools*,  kemudian Klik pada blok pertama yang kita buat (alamat 000.00), setelah terlihat gelap klik pada *vertical line* dan klik pada blok kedua baris pertama sehingga akan timbul garis penghubung ke bawah.



14. Klik  pada toolbar *symbol* dan Klik pada *work area* baris kedua blok pertama, kemudian masukan alamat / *adres* yang sama dengan internal relay, 001.00
15. Klik kembali  pada *toolbar symbol* dan Klik pada *work area* baris ketiga blok pertama, kemudian masukan alamat / *adres* yang sama dengan *internal relay*, 001.00 .
16. Klik *symbol*  atau *open output* pada *toolbar symbol* pada *work area* pada baris ketiga blok ketiga, dan berikan alamat
17. Tempatkan pada area kerja, sehingga akan muncul dialog form dialog.

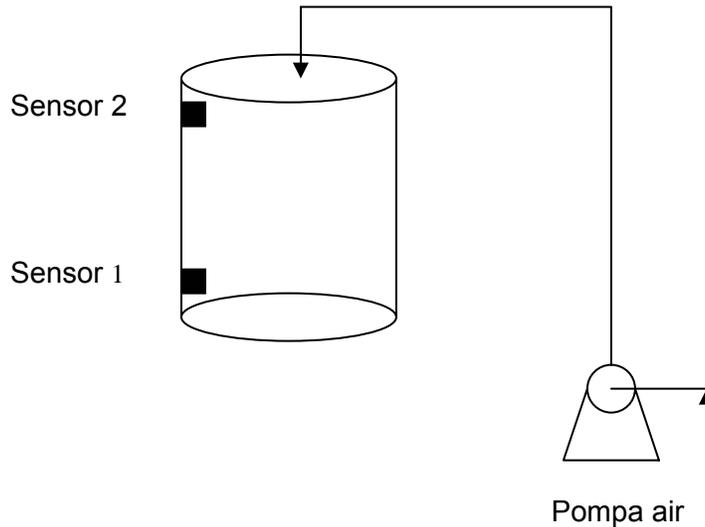


18. Hubungkan blok pertama dan ketiga pada baris ketiga dengan garis horizontal  pada *toolbar symbol*.
19. masukan *Function END* pada bris terakhir.

3.18.3 Pengisian tangki air otomatis

Sistem pengisian tangki air secara otomatis dapat dijelaskan pada gambar di bawah ini. Awalnya tangki dalam keadaan kosong, pompa air hidup sehingga mulai mengisi air dari keadaan awal kosong. Saat air menyentuh sensor 2 pompa air masih hidup dan pengisian air terus berlanjut sampai sensor 1 . Ketika sensor 1 tercelup oleh air pompa berhenti sehingga volume air secara perlahan turun. Beberapa saat kemudian

sensor air volume air terus menurun sehingga kini sensor 1 dalam keadaan bebas atau tidak tercelup air dan pompa masih terus hidup. Sampai ketinggian air menyentuh sensor 2 pompa kembali menyala dan terus berulang.



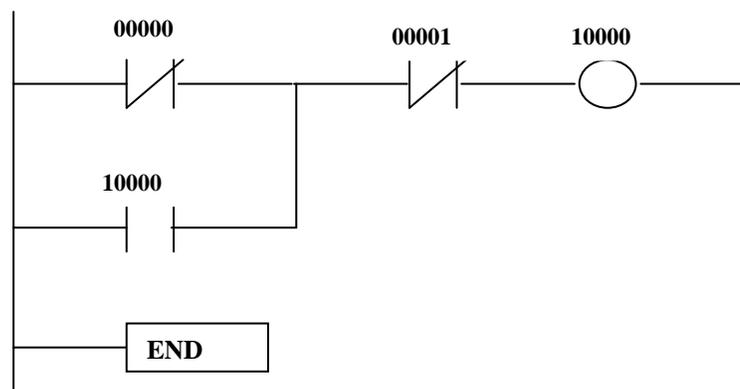
Gambar 47. Sistem kerja pengisian air otomatis

Langkah selanjutnya adalah membagi alamat *input* dan *output* pada *Programmable Logic Control* (PLC). Perlu diperhatikan bahwa sistem di atas memiliki *input dan output (i/o)* sebanyak tiga bagian dengan dua *input* dan satu *output*. Input terdiri dari sensor 1 dan sensor 2, sedangkan output berupa pompa air.

Tabel 6. Daftar alamat pada *input* dan *output*

Input	Adres	Output	Adres	Internal relay
Sensor 1	000.00	Pompa air	001.00	010.00
Sensor 2	000.01			

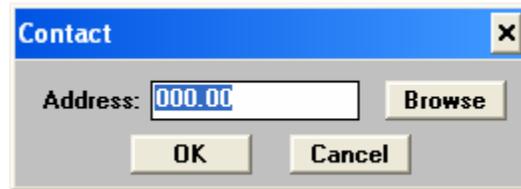
Setelah dibuat alamat pada *input* dan *output* PLC selanjutnya adalah membuat diagram ladder. Pada kasus di atas menggunakan alamat-alamat PLC jenis OMRON.



Gambar 48. Diagram ladder

Contoh pembuatan diagram ladder pada program computer jenis syswin 3.4

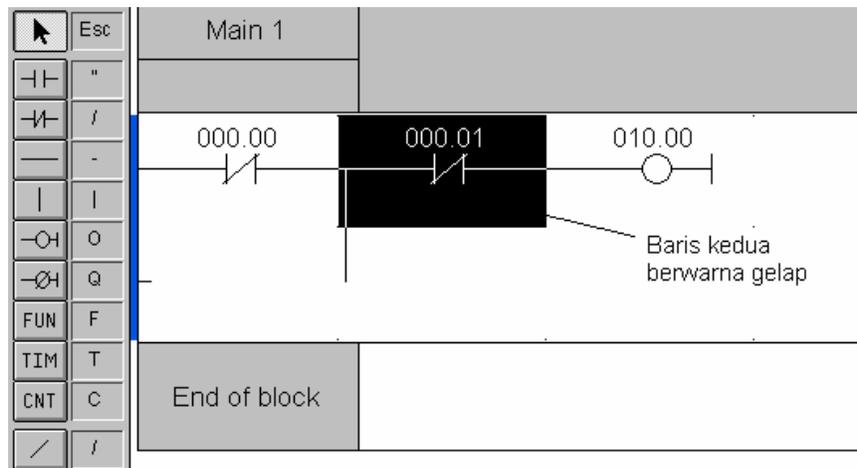
1. Klik gambar *closed contact*  pada sisi kiri yaitu *toolbar* simbol.
2. Tempatkan pada *work area*, kemudian muncul *form dialog box* untuk pengisian *adres / alamat*



3. Isikan alamat untuk sensor atas
4. Klik kembali *closed contact*  pada *toolbar* simbol dan Klik pada *work area* samping ladder pertama, kemudian masukan alamat / *adres* yang berbeda dengan *address* sensor .
5. Klik *symbol*  atau *open output* pada *toolbar symbol* sebelah ladder kedua. Tempatkan pada area kerja, sehingga akan muncul *form dialog*.



6. Isikan alamat *output*, sesuai dengan ladder yang telah kita tentukan yaitu 010.00
7. Klik pada *selection tools*,  kemudian Klik pada blok kedua yang kita buat (alamat 000.01), setelah terlihat gelap klik pada *vertical line* dan klik pada blok kedua baris pertama sehingga akan timbul garis penghubung ke bawah.



8. Klik gambar *open contact*  pada sisi kiri yaitu *toolbar simbol*.
9. Tempatkan pada *work area* baris kedua blok pertama kemudian muncul form dialog bok untuk pengisian *adres / alamat*



10. Isikan alamat yang sama dengan *output* yaitu 01.00. *open contact* disini berfungsi sebagai *internal relay* sebagai proses *latching* atau *self holding*.
11. masukan *Function END* pada bris terakhir.

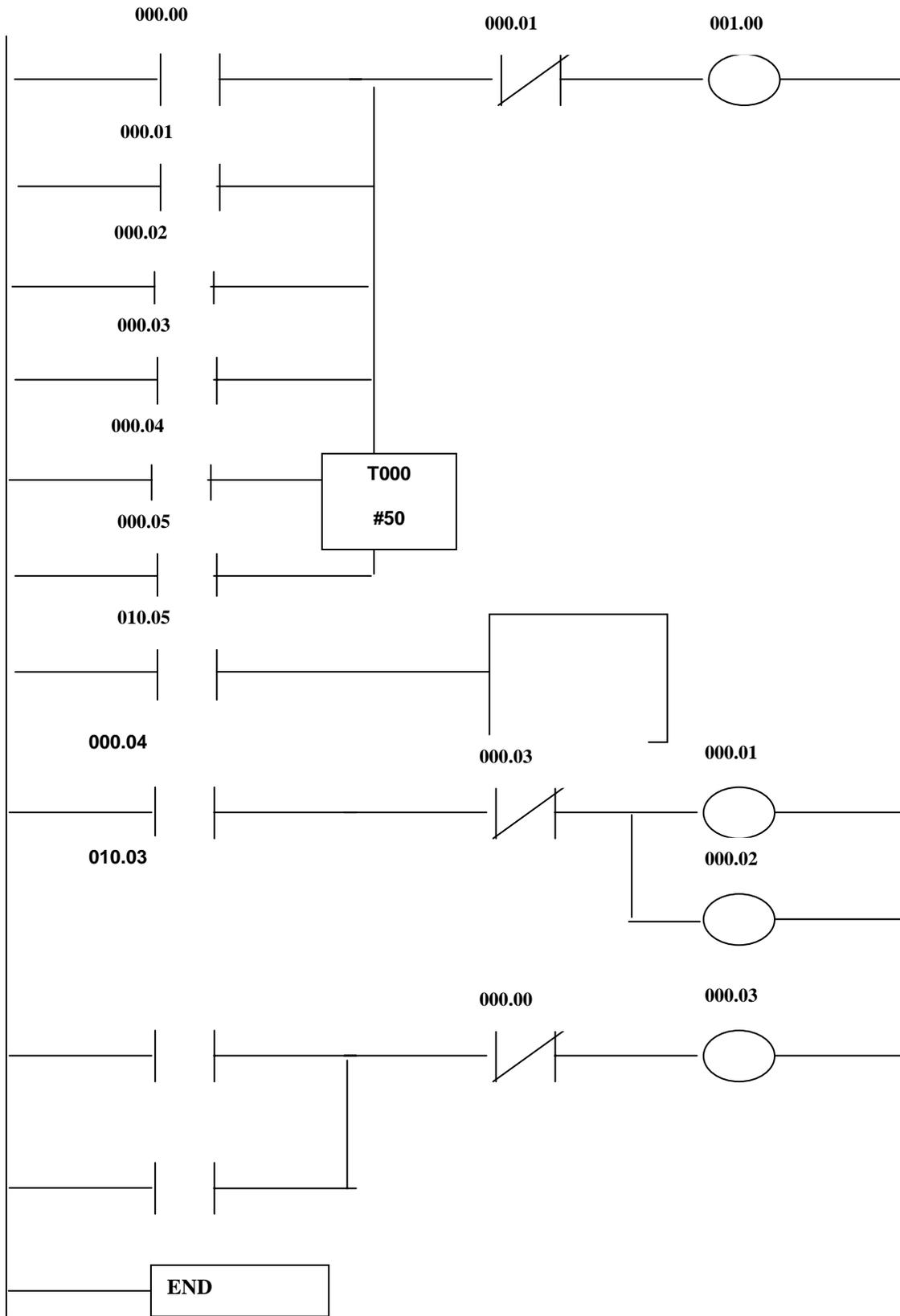
3.18.4 Alarm dan sistem penyalan lampu

Suatu pabrik tiap pukul 07.30 membunyikan alarm yang menandakan karyawan untuk masuk bekerja bersamaan dengan alarm yang berbunyi lampu dalam kantor dan lampu ruang kerja secara otomatis akan menyala dan lampu lingkungan perusahaan akan padam. Kemudian pada pukul 11.30 alarm berbunyi untuk waktu istirahat dan pukul 12.30 alarm berbunyi tanda masuk kembali. Alarm akan berbunyi lagi pada pukup 16.30 sebagai tanda jam bekerja telah berakhir. Lampu lingkungan perusahaan akan menyala pukul 17.30.

Table 7. Alamat *input* dan *output* PLC

Input	Alamat	Output	Alamat
Jam 07.30	000.00	Alarm	010.00
Jam 11.30	000.01	Lampu ruang kantor	010.01
Jam 12.30	000.02	Lampu ruang kerja	010.02
Jam 16.30	000.03	Lampu lingkungan	010.03
Jam 17.30	000.04		010.04

Pada tabel di atas kita menjadi mengerti alamat masing masing *input* dan *output* PLC. Alamat input di tentukan dengan waktu dan alamat output di hubungkan dengan lampu dan alarm. Pengaplikasian pada program syswin 3.4 dapat disesuaikan dengan contoh program syswin di atas.



Gambar 49. Diagram Ladder aplikasi sistem alarm

DAFTAR PUSTAKA

- Delman kilian, **Modern Control Technology Component And System Handbook**, Omega Engineering, Inc, Stamford 1999
- Depdikbud, 1995, **Mesin Bubut CNC Dasar**, Jakarta.
- J.J.M. Hollebrandse, Soedjono. 1988. **Teknik Pemrograman Dan Aplikasi CNC**. Jakarta: PT Rosda Jayaputra.
- Katsuhiko Ogata : **Teknik Kontrol Automatik (Sistem Pengaturan Jilid 1)** ; Penerbit Erlangga ; Jakarta
- Keller, 1992, **Schulungsunterlagen CNC Maho 432**, Solingen, CNC Didaktik.
- Lilih Dwi P., 2001, **Buku CNC Milling – TU 2A (Mesin Bubut Dasar)** , Laboratorium CNC – BLPT Surabaya.
- Lilih Dwi P., 2001, **Buku CNC Milling – TU 3A (Mesin Freis Dasar)** , Laboratorium CNC – BLPT Surabaya.
- Maier dan Co, 1988, **Pelayanan EMCO 2A**, Hallem, EMCO
- Maier dan Co, 1988, **Pelayanan EMCO 3A**, Hallem, EMCO
- Meier (1992), **Petunjuk Penggunaan Mesin CNC EMCO TU-3A**, Austria, EMCO
- Mikell P. Groover: **Automation Production systems, and Computer-Integrated Manufacturing** : Pearson Education ; Singapore, 2001
- Pakpahan : **Kontrol Otomatik** ; Penerbit Erlangga ; Jakarta, 1984
- Sachur Rheinhard, 1988, **CNC Technik**, Homburg, Gehlen
- Setiawan, Iwan. 2006. **Programmable Logic Controller (PLC) dan Perancangan Sistem Kontrol**. Yogyakarta : Penerbit Andi
- Soewito, Hadi. 1992. **Pengetahuan Dasar Mesin CNC**. Bandung: Pusat Pengembangan Penataran Guru Teknologi Bandung.
- Sumbodo, Wirawan, 1998, **Dasar-dasar Sistem Pemrogramman mesin CNC**, Semarang, Unnes
- [www.Wikipedia.com/id/search/automation system](http://www.Wikipedia.com/id/search/automation%20system)
- www.en.wikipedia.com
- www.cnc-keller.de

www.hondacompany.com

[www.Q Plus Frasen.com](http://www.QPlusFrasen.com)

www.omron.com/index3.html

www.zen.omron.co.jp/eng/index.html - 22k

www.plcs.net/ - 20k

www.automation.com

world.honda.com/ASIMO/