

Modul Perkuliahan

Teknik Mesin UMY

Otomasi dan PLC

Bambang Riyanta

PENGANTAR

Assalamu'alaikum wr.wb.

Mata Kuliah Otomasi dan PLC adalah mata kuliah wajib 2 sks dengan kode mata kuliah TMU 563 pada semester 4 di Program Studi Teknik Mesin.

Learning Outcome yang diharapkan dari Mata Kuliah ini adalah :

Mampu memahami, menggunakan dan merangkai sistem pengendalian hidrolik, pneumatik dan elektropneumatik sesuai dengan kebutuhan serta menggunakan program ladder untuk mendukung pemanfaatan PLC

Diktat OTOMASI DAN PLC disusun dalam rangka mendukung pencapaian kompetensi yang diharapkan pada mata kuliah Otomasi dan PLC

Banyak kekurangan yang ada pada diktat sederhana ini, oleh karenanya masukan dari pembaca akan sangat berguna bagi penyempurnaan diktat kuliah ini.

Wassalamualaikum wr.wb.

Yogyakarta, Februari 2016

Bambang Riyanta, S.T.,M.T.

DAFTAR ISI

Pengantar

Daftar isi

BAB I.	SISTEM HIDROLIK	4
BAB II.	SISTEM PNEUMATIK	9
BAB III	SISTEM ELEKTROPNEUMATIK	27
BAB IV	PROGRAMABLE LOGIC CONTROL	42

Daftar Pustaka

BAB I

SISTEM HIDROLIK



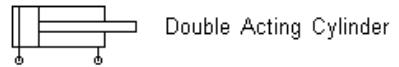
I.1. SISTEM HIDROLIK :

Prinsip dasar dari hidrolik adalah pemindahan tenaga dengan menggunakan fluida sebagai perantara . Sifat fluida cair yang tidak termanfaatkan, meneruskan tekanan ke segala arah, mengikuti bentuk wadah serta meneruskan tekanan ke segala arah dengan sama rata memberikan keuntungan pada system hidrolis berupa : gerakan yang halus, mampu menerima beban kejut, pelaksanaan kerja otomatis, konstruksi sederhana, keausan rendah, rentang kecepatan dan pembebanan besar.



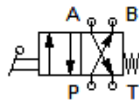
I.2. KOMPONEN SISTEM HIDROLIK

1. Cylinder

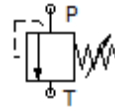


Double Acting Cylinder

2. Valve

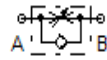


4/2 Way Hand Lever Valve



Pressure Relief Valve

3. Flow Control



One Way Flow Control Valve



2- Way Flow Control Valve

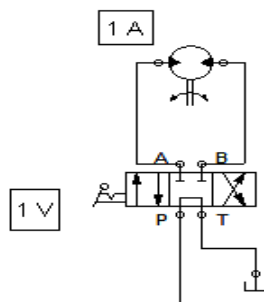
4. Pompa



 Pump Unit

LATIHAN

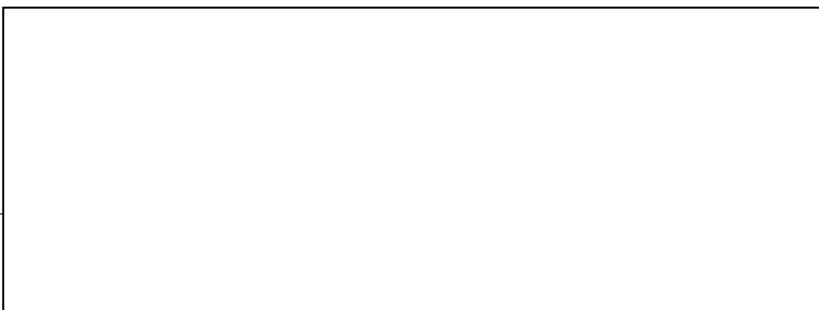
1. Sebuah mesin pengaduk digerakan dengan sistem hidrolik. Poros pengaduk dapat berputar searah jarum jam, berlawanan jarum jam ataupun dapat dihentikan tanpa menghentikan mesin. Pengaduk dioperasikan dengan satu tombol operator.
 - Apa saja komponen yang dibutuhkan untuk sirkuit hidrolik tersebut ?
 - Buatlah rangkaian sirkuit tersebut menggunakan software FluidSIM Hidrolik !
 - Buatlah rangkaian sirkuit tersebut di meja peraga !



Gambar 1. Rangkaian menggunakan software FluidSIM Hidrolik

2. Sebuah penutup mesin digerakan dengan sistem hidrolik. Penutup membuka jika operator menggerakan tombol. Jika tombol dilepas, maka penutup mesin tersebut akan menutup dengan sendirinya. Jika tekanan melebihi batas tertentu, aliran dari pompa akan mengalir kembali ke tangki.
 - Apa saja komponen yang dibutuhkan untuk sirkuit hidrolik tersebut ?
 - Buatlah rangkaian sirkuit tersebut menggunakan software FluidSIM Hidrolik !
 - Buatlah rangkaian sirkuit tersebut di meja peraga

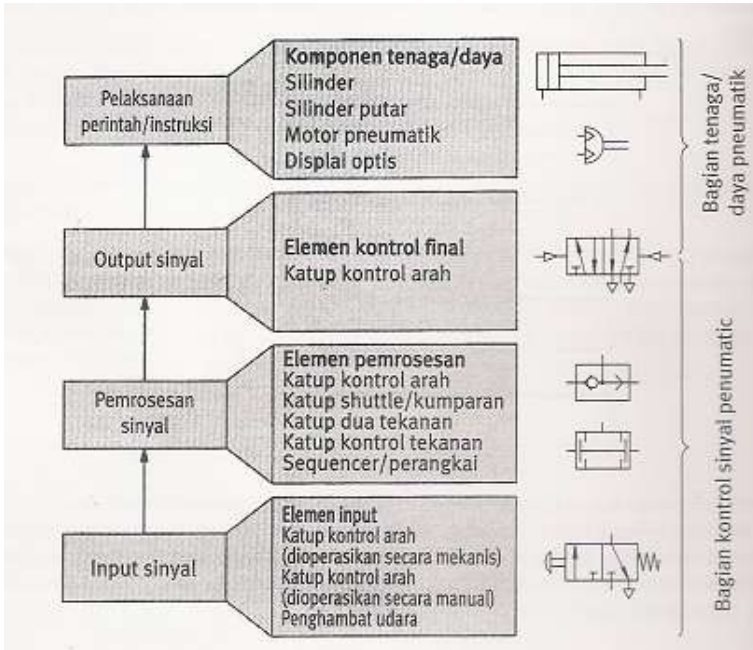
Gambar Rangkaian sirkuit menggunakan software FluidSIM – H



BAB II SISTEM PNEUMATIK

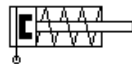


II.1. STRUKTUR ALIRAN SINYAL SISTEM PNEUMATIK

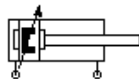


II.2. KOMPONEN SISTEM PNEUMATIK

1. Cylinder

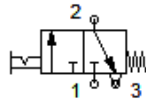


Single Acting Cylinder

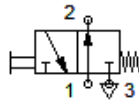


Double Acting Cylinder

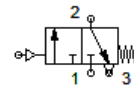
2. Valve



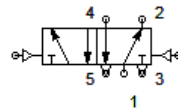
3/2 Way Valve as Detach



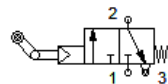
3/2 Way Valve as Push Button



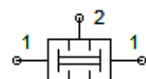
3/2 Way Valve



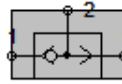
5/2 Way Valve



Limit Switch



Two pressure Valve



Shuttle Valve

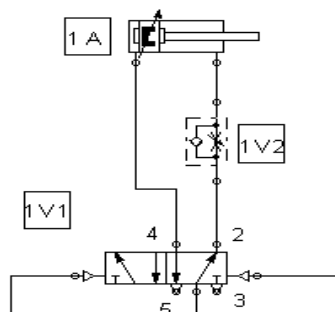
3. Flow Control



One Way Flow Control Valve

LATIHAN

1. Sebuah pintu air digerakan dengan sistem pneumatic. Pintu terbuka ketika tombol pembuka ditekan dan pintu tertutup jika tombol penutup ditekan. Kecepatan pintu membuka dapat diatur.
 - Apa saja komponen yang dibutuhkan untuk sirkuit Pneumatic tersebut ?
 - Buatlah rangkaian sirkuit tersebut menggunakan software FluidSIM Pneumatic !
 - Buatlah rangkaian sirkuit tersebut di meja peraga !



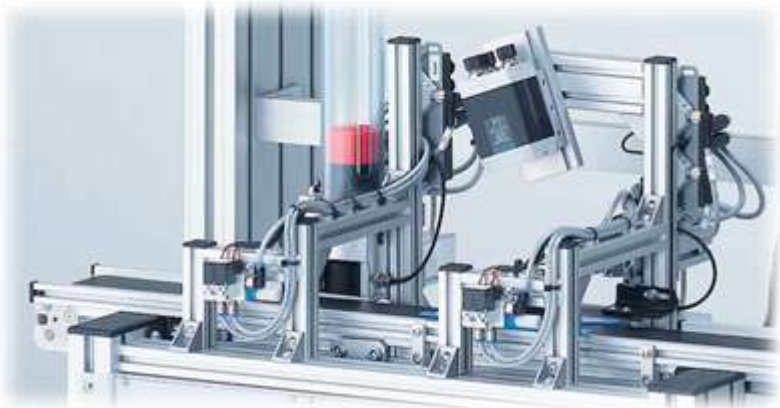
Gambar 2. Rangkaian menggunakan software FluidSIM Pneumatic

2. Sebuah mesin pelubang plat digerakan dengan tenaga pneumatic. Untuk proses pembuatan lubang, pisau pelubang bergerak maju jika tombol ditekan oleh operator. Pisau bergerak mundur keposisi awal jika tombol dilepas oleh operator. Karena berat beban, maka harus menggunakan kontrol tidak langsung.
 - Apa saja komponen yang dibutuhkan untuk sirkuit Pneumatic tersebut ?
 - Buatlah rangkaian sirkuit tersebut menggunakan software FluidSIM Pneumatic !
 - Buatlah rangkaian sirkuit tersebut di meja peraga

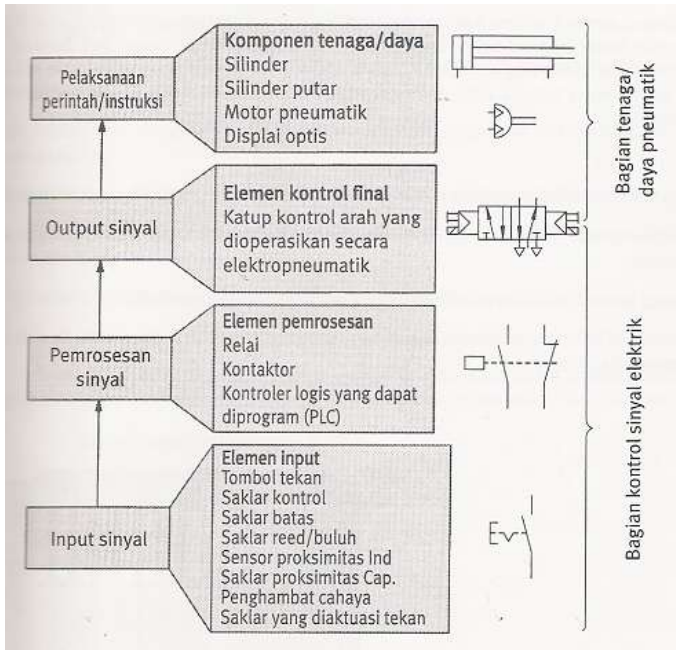


Gambar Rangkaian sirkuit menggunakan software FluidSIM – P

BAB III SISTEM ELECTROPNEUMATIK

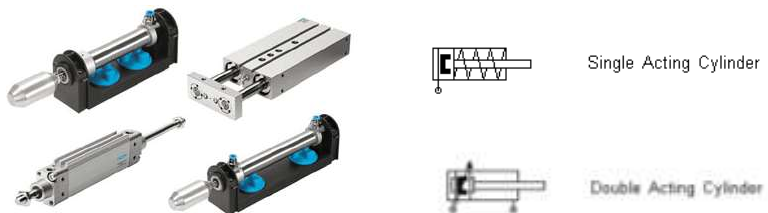


III.1. STRUKTUR ALIRAN SINYAL SISTEM ELECTROPNEUMATIK



III.2. KOMPONEN SISTEM ELECTROPNEUMATIK

1. Cylinder

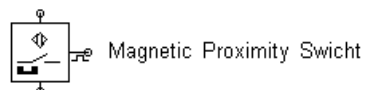


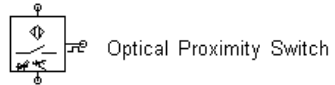
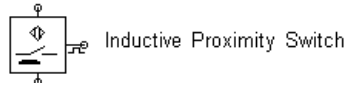
2. Solenoid Valve



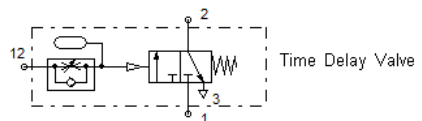
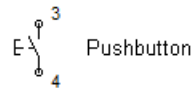
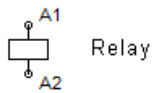


3. Proximity Sensor





4. Relay , Swicth, Timer



III.3. SYMBOL ACTUATOR

Fungsi	Uraian	Simbol
Silinder daya tunggal	Maju dengan tenaga pneumatik Mundur dengan pegas balik	
Silinder daya ganda	maju dan mundur dengan tenaga pneumatik	
Silinder daya ganda	Bantalan posisi ujung yang dapat disetel untuk ayunan maju dan mundur	
Silinder daya ganda dengan unit penjepit	Unit penjepit mekanis dengan pneumatik yang tidak mengunci	
Silinder daya ganda dengan silinder slave* hidrolis	Silinder dikendalikan secara pneumatik Silinder slave hidrolis memberikan suatu pergerakan yang sama/merata	
Silinder tanpa batang dengan bantalan posisi ujung yang dapat disetel	Biasanya silinder dengan kopanjang ayunan yang panjang. Transmisi tenaga dengan magnet permanen	
Silinder tanpa batang dengan bantalan posisi ujung yang dapat disetel	Transmisi tenaga/daya dengan alat mekanis	
Penggerak vane*, pneumatik	Penggerak putar dengan kisaran putar terbatas	
Motor udara terkompresi, pneumatik	Motor udara terkompresi dengan kapasitas tetap dan satu arah putar.	
Motor udara terkompresi.	Motor pneumatik dengan dua arah putar	
Generator vacuum	Input vacuum melalui ejector	

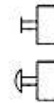
Informasi berikut ini diperlukan untuk secara lengkap merupakan suatu katup kontrol arah dalam diagram circuit pneumatik:

- Jenis dasar dari aktuasi katup
- Metode pemasangan kembali
- Kontrol pilot (apabila dapat diterapkan)
- Bentuk-bentuk aktuasi tambahan (seperti penolakan/override manual, jika ada)

Masing-masing simbol aktuasi digambarkan pada sisi posisi saklar yang berpadanan dengan arah tindakan.

Operasi manual

- Umum
- Dengan tombol tekan



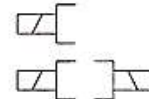
Pemasangan kembali secara mekanik

- Dengan pegas
- Pegas yang terpusat



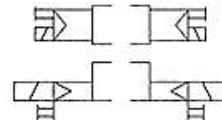
Aktuasi listrik

- Dengan satu solenoida
- Dengan dua solenoida



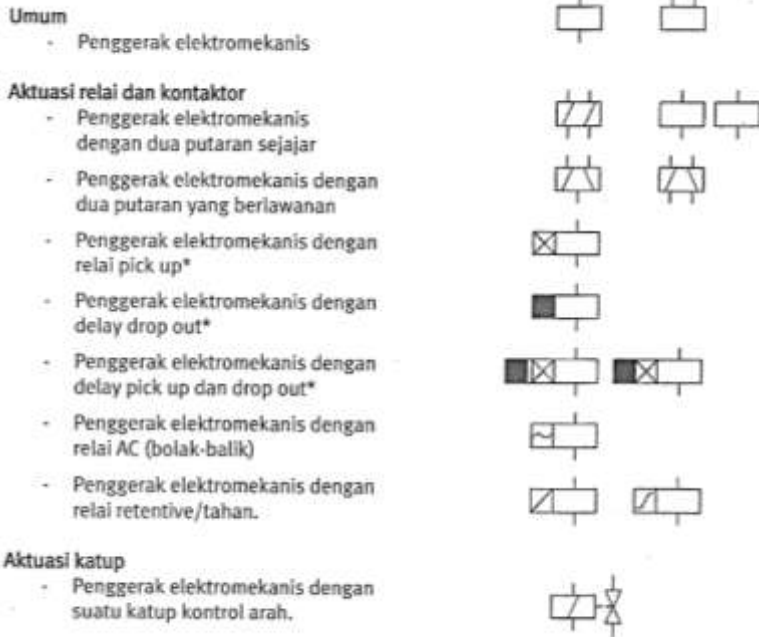
Aktuasi kombinasi

- Dengan katup yang diaktuasikan oleh pilot, solenoida ganda, override manual/ tolakan manual

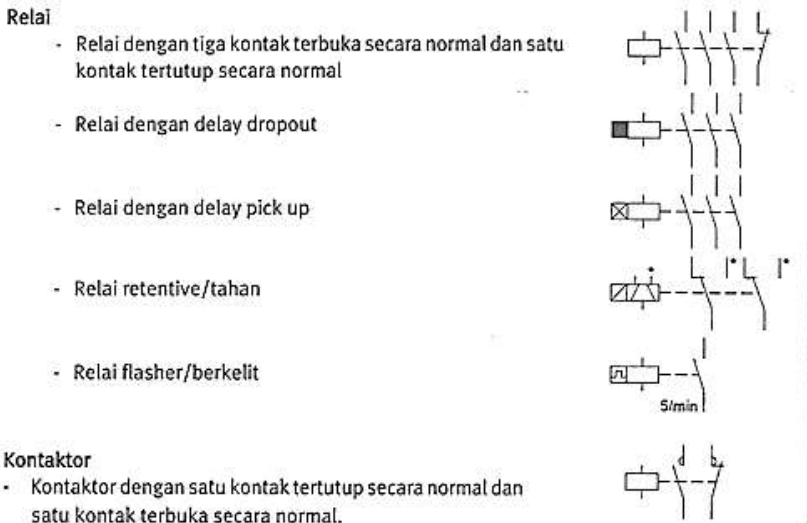


III.5. SAKLAR DAN PENGGERAK

	Saklar kontak momentary/sebentar	Saklar jenis latching/pegas
Kontak terbuka secara normal dioperasikan secara manual		
Kontak terbuka secara normal dioperasikan secara manual dengan menekan		
Kontak tertutup secara normal ioperasikan secara manual dengan menarik		
Kontak tertutup secara normal dioperasikan secara manual dengan memutar		



III.6.. RELAI



Ketika suatu voltase diterapkan pada terminasi getar yang ditandai dengan *, label kontak ditandai dengan posisi elemen kontak yang ditandai dengan *.

III.7. SENSOR

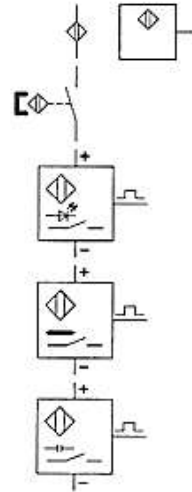
Saklar batas

- Kontak terbuka secara normal
- Kontak tertutup secara normal



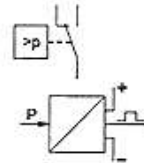
Saklar proximitas, sensor proximitas

- Sensor proximitas, alat peka proximitas
- Saklar proximitas (kontak terbuka secara normal) yang diaktivasikan oleh magnet (terstandarisasi).
- Saklar proximitas, optis
- Saklar proximitas, induktif
- Saklar proximitas, kapasitif



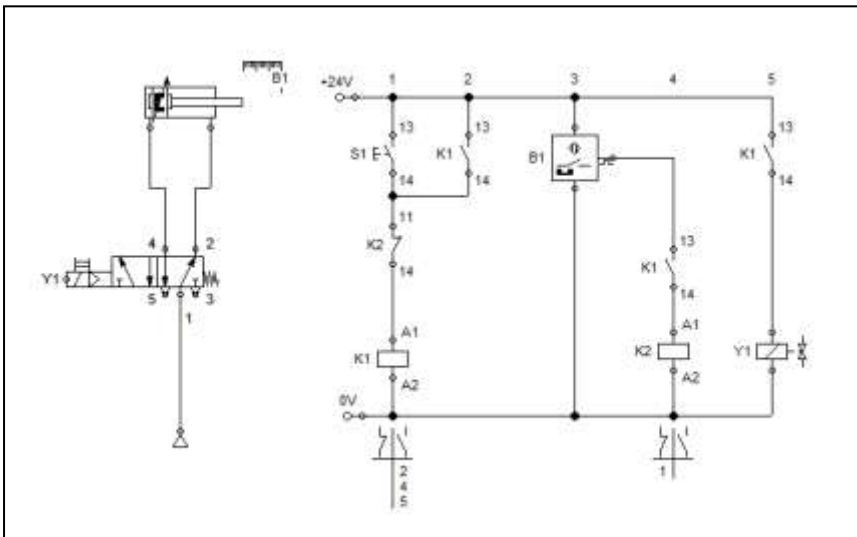
Saklar tekan, sensor tekan

- Saklar tekan, elektromekanis
- Saklar tekan, elektronik



LATIHAN

1. Sebuah benda didorong keluar ke atas ban berjalan dengan menggunakan silinder kerja ganda. Pekerjaan untuk mulai diberikan dengan menekan tombol. Apabila silinder keluar sudah mencapai posisi maksimum, silinder kembali secara otomatis keposisi semula. Untuk menjamin bahwa benda benar-benar terletak diatas banberjalan diperlukan sebuah sinyal.
 - Apa saja komponen yang dibutuhkan untuk sirkuit ElektroPneumatic tersebut ?
 - Buatlah rangkaian sirkuit tersebut menggunakan software FluidSIM Pneumatic !
 - Buatlah rangkaian sirkuit tersebut di meja peraga !



Gambar 3. Rangkaian menggunakan software FluidSIM Pneumatic

2. Dengan menggunakan alat pemindah jalur ban berjalan, benda diatas kerangka alat tersebut dipindah dari ban satu ke ban lain. Kerangka pemindah dapat maju jika tombol switch (S1) ditekan. Benda pindah dari ban yang satu ke ban lain. Untuk mengembalikan kerangka pemindah keposisi semula harus menekan tombol yang lain (S2). Jalur lintasan terhubung sesuai perintah terakhir yang diberikan.
 - Apa saja komponen yang dibutuhkan untuk sirkuit ElektroPneumatic tersebut ?
 - Buatlah rangkaian sirkuit tersebut menggunakan software FluidSIM Pneumatic !

Buatlah rangkaian sirkuit tersebut di meja peraga

Gambar Rangkaian sirkuit menggunakan software FluidSIM - P



BAB IV

PROGRAMMABLE LOGIC CONTROL (PLC)

Jika berkunjung ke suatu pabrik dan diberi kesempatan untuk melihat sistem pengendalian motor-motor induksi yang digunakan untuk menggerakkan mesin atau conveyor, maka tidak jarang menemui sebuah kotak (mirip panel box listrik) yang dilengkapi dengan saklar dan lampu indikator, kotak panel itulah yang disebut dengan panel kontrol (*control panel*).

Saklar digunakan untuk menghidupkan atau mematikan motor, lampu, kipas, dan lain sebagainya, saklar tersebut tidak langsung dihubungkan ke obyek yang akan dikendalikan, tetapi digunakan untuk mengendalikan relay, relay inilah yang menghidupkan atau mematikan suatu obyek (motor, alarm, dll). Pada panel kendali tersebut terdapat banyak titik keluaran yang dihubungkan ke obyek yang dikendalikan dengan menggunakan kabel, sehingga tak heran jika ukuran kotak panel menjadi besar,. Disamping relay dan tombol, isi sebuah panel kendali terdapat pula counter, dan timer.

PLC (Programable logic controller) dibuat untuk menggantikan fungsi control panel, isi panel kendali tergantung kebutuhan, sedangkan *PLC* dibuat seragam, untuk menyesuaikan dengan kebutuhan, *PLC* dilengkapi dengan program, dengan kata lain *PLC* digunakan untuk menggantikan fungsi panel kendali namun sangat fleksibel.

PLC mempunyai kepanjangan *Programable Logic Controller*, atau pengendali logika yang terprogram. Pada awal pembuatan, *PLC* merupakan pengganti panel listrik yang terdiri atas counter, timer, relay, saklar, dan lain sebagainya, bedanya hanya pada kemampuannya untuk dapat diprogram.

Meskipun *PLC* dibuat untuk menggantikan sebuah panel kendali, tetapi kebutuhan kian meningkat, fungsi sebuah *PLC* bergeser hingga mirip fungsi sebuah komputer mikro tetapi hanya khusus untuk pengendalian. Saat ini bahkan *PLC* menjadi peralatan utama dalam sistem kendali terdistribusi

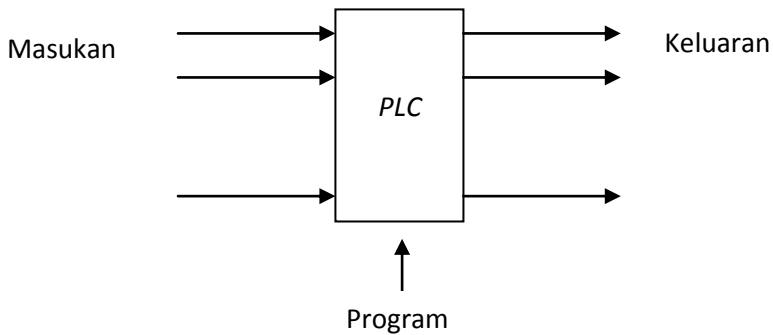
(Distributed control system). *PLC* sudah bisa melakukan pengolahan secara numeris sederhana, dilengkapi dengan peralatan komunikasi sehingga bisa membentuk jaringan.

BAB ini disusun untuk memberikan informasi tentang *PLC*, secara praktis. Kemampuan menggunakan *PLC* merupakan modal bagi mahasiswa teknik untuk menyongsong era industri *modern*, Kompetensi otomasi dan *PLC* dapat diperoleh hanya dengan pemahaman teori dari perkuliahan serta latihan yang intensif melalui pelatihan, praktikum, atau terjun langsung secara profesional.

PLC di industri digunakan sebagai peralatan kontrol yang dapat diprogram untuk mengontrol proses atau operasi mesin. Kontrol program dari *PLC* menganalisa sinyal input kemudian mengatur keadaan output sesuai dengan keinginan pemakai. Input *PLC* disimpan didalam memory. Sesuai program yang telah berada pada memori, *PLC* akan bekerja dengan operasi logika atau sequencial . Peralatan input dapat berupa sensor *photo elektrik*, *push button* pada panel kontrol, *limit switch*. Peralatan *output* dapat berupa *switch* yang menyalakan lampu indikator, *relay* yang menggerakkan motor atau peralatan lain yang dapat digerakkan oleh sinyal *output* dari *PLC*. Selain itu *PLC* juga menggunakan *memory* yang dapat diprogram untuk menyimpan instruksi – instruksi yang melaksanakan fungsi – fungsi khusus seperti : logika pewaktuan, sekuensial dan aritmetika yang dapat mengendalikan suatu mesin atau proses melalui modul – modul I/O baik analog maupun digital

IV.1. MENGENAL PLC

Suatu *Programmable Logic Controller (PLC)* pada mulanya adalah suatu CPU yang berisi suatu program dan dihubungkan dengan peralatan *input / output (I/O)* yang berupa relay (*solid state relay*). Pada dasarnya sebuah *PLC* mempunyai sederet masukan yang dihubungkan dengan sensor, saklar, atau peralatan lain yang dapat memberikan isyarat (baik analog maupun digital, selain itu *PLC* mempunyai sederet keluaran yang dihubungkan dengan peralatan luar pula yang dapat menerima isyarat dari *PLC*, misal lampu, relay, transistor, dll.



Gambar 2.1 Deret PLC

Jadi, masukan mempunyai hubungan secara logika terhadap keluaran tergantung dari logika yang diberikan pada *PLC*, Logika tersebut dapat diprogram. Oleh karena itulah alat ini diberi nama logika pengendali terprogram atau *PLC (programmable logic controller)* Selain melalui pengolahan masukan secara logika, saat ini *PLC* dapat berfungsi untuk mengolah isyarat masukan baik digital maupun analog dan memproses secara numeris sederhana (layaknya sebuah operasi sederhana pada komputer mikro).

Program yang sudah diberikan akan mengendalikan *PLC*, antara masukan dan keluaran mempunyai hubungan secara logika, sehingga bila ada suatu peralatan masukan memberi isyarat, maka *PLC* akan menanggapi dan

memproses isyarat tersebut dan memberikan suatu keputusan yang biasanya berupa isyarat pada terminal keluarannya.

Peralatan *input* dapat berupa sensor cahaya (*photoelectric*), tombol pada panel, saklar pembatas (*limit switches*), atau setiap peralatan *input* yang bisa memberikan isyarat masukan pada *PLC*. Sedangkan peralatan *output* dapat berupa *solenoid*, saklar yang mengaktifkan lampu, *relay* untuk mengaktifkan sebuah motor atau setiap peralatan yang dapat digerakan oleh isyarat *output* *PLC*.

Pada awalnya perkembangan *PLC* adalah suatu sistem kendali yang berbasis *relay*. Sistem ini merupakan untai terintegrasi yang mengambil alih tugas-tugas peralatan seperti *relay*, *timer* dan *counter*. Saat ini *PLC* mempunyai kemampuan seperti komputer dan mempunyai fleksibilitas dan realibilitas yang lebih baik di bandingkan dengan sistem *relay*. Simbol-simbol yang digunakan dalam *PLC* berbasis *relay*, tetapi sebagian besar istilah untuk menjelaskan simbol dan konsep datang dari istilah komputer seperti *and*, *or*, *not*, *jump*, dan sebagainya. Disamping itu saat ini *PLC* mempunyai masukan analog, demikian juga halnya dengan keluarannya.

IV.2. KEUNTUNGAN MENGGUNAKAN PLC

Pada dasarnya *PLC* digunakan untuk mengganti sistem kontrol panel yang telah lama dipakai, *PLC* dengan konfigurasi dasar, mempunyai jenis masukan / keluaran (I/O) berupa sinyal logic *ON-OFF* (1 dan 0). Prinsip kerja dari *Programmable Logic Controller (PLC)* sama dengan prinsip kerja *relay* yaitu berupa saklar *ON-OFF*. Namun saat ini fungsi *PLC* sudah jauh dari fungsi tersebut yaitu dilengkapi dengan peralatan analog, jaringan *PLC* yang dapat diintegrasikan dengan jaringan komputer. Keuntungan yang bisa diperoleh jika menggunakan *Programmable Logic Controller (PLC)* dibanding dengan kontrol panel :

1. Perkawatan (*wiring*) lebih sedikit
2. Komponen cadangan mudah di dapat
3. Pemeliharaan lebih mudah

4. Ukuran jauh lebih kecil, sebagai gambaran besamya PLC mungkin lebih kecil dari relay yang digunakan pada kontrol panel biasa.
5. Disain baru lebih mudah direalisasikan
6. Fungsi dapat disesuaikan dengan kebutuhan dengan penambahan sedikit peralatan.
7. Dilengkapi dengan sistem pelacakan kesalahan
8. Konsumsi daya sangat rendah.
9. Dokumentasi lebih mudah, sederhana, dan mudah dimengerti
10. Dapat diintegrasikan dengan PLC yang lain dalam bentuk jaringan PLC, maupun jaringan komputer.
11. Pengubahan kerja sistem lebih mudah oleh karena berbentuk program.
12. Kemudahan untuk memprogram dan mengubah program.
13. Kemudahan dalam pemeliharaan dan perbaikan.
14. Dapat digunakan pada bermacam-macam alat dan ukurannya lebih kecil daripada *relay* pada umumnya.

IV.3. KONSEP DASAR LOGIKA TERPROGRAM

Hal mendasar yang harus diketahui dalam mengenal dan menggunakan *Programmable Logic Controller (PLC)* sesuai fungsi yang diinginkan adalah:

IV.3.1. Prinsip Kerja

Programmable Logic Controller (PLC) tersusun dari seperangkat peralatan elektronik yang dilengkapi *mikroprosesor* untuk memonitor keadaan dari peralatan *input* guna mengendalikan keadaan *output* sesuai dengan kebutuhan perencana (*programmer*). Sinyal *input* diberikan kedalam *input card*. Ada 2 jenis *input card*, yaitu :

1. *Analog input card*
2. *Digital input card*

Setiap *input* mempunyai alamat tertentu . *Mikroprosesor* memanggil berdasarkan alamatnya. Banyaknya *input* yang dapat diproses tergantung jenis *PLC*. Sinyal *output* dikeluarkan *PLC* sesuai dengan program yang dibuat oleh pemakai berdasarkan analisa keadan *input*. Ada 2 jenis *output card*, yaitu :

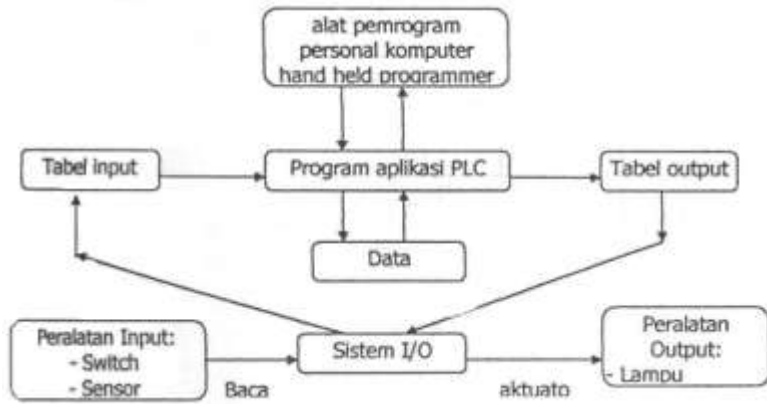
1. *analog output card*
2. *digital output card*

Setiap *output card* mempunyai alamat tertentu dan diproses oleh *mikroprosesor* menurut alamatnya. Banyaknya *output* tergantung jenis *PLC*. Pada *PLC* juga terdapat internal *input* dan *output* untuk keperluan proses di dalam *PLC* sesuai dengan kebutuhan program. Di dalam *PLC* juga dipersiapkan *timer* yang dapat disetting dalam konfigurasi *on delay* ,*off delay*, *ontimer*, *off timer* dan lain- lain sesuai dengan programnya. Ketika diperlukan, *PLC* akan memanggil timer berdasarkan alamatnya.

PLC didukung oleh perangkat lunak. Program *PLC* biasanya terdiri dari 2 jenis yaitu *ladder* diagram dan *instruksi* dasar diagram, setiap *PLC* mempunyai perbedaan dalam penulisan program.

PLC menerima sinyal Input dari peralatan diskrit (*ON / OFF*) atau *analog(Sensor)*. Modul *input* mendeteksi serta mengubah sinyal tersebut kedalam bentuk tegangan yang sesuai dan mengirimnya ke *CPU (Cental Processing Unit)*. Sinyal masukan tersebut kemudian diolah secara logika ataupun secara *numerus* dan dikirim ke modul keluaran berdasarkan program yang tersimpan di *CPU (Central Processing Unit)*, bentuk sinyal keluaran diubah menjadi tegangan yang sesuai dan dipakai untuk menjalankan peralatan keluaran (*aktuator*).

Dengan *blok diagram* dapat digambarkan seperti terlihat pada gambar 2.1 berikut ini :



Gambar 2.2. Blok Diagram Prinsip Kerja PLC

IV.3.2. Sistem Alur program

Pada diagram *ladder* sinyal mirip arus yang mengalir pada rangkaian listrik yang mengalir dari kiri ke kanan. Garis vertikal pada posisi kiri adalah rel sumber daya untuk mengaktifkan fungsi-fungsi yang terdapat dalam program yang dibuat.

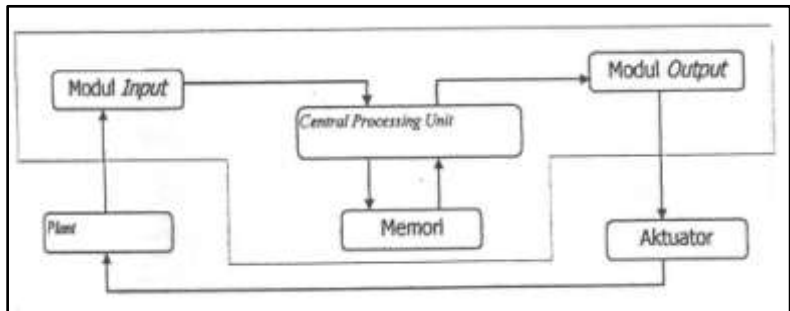


Gambar 2.3. Ladder diagram PLC

IV.3.3. Perangkat Keras *Programmable Logic Controller (PLC)*

Seperti telah dijelaskan sebelumnya, *Programmable Logic Controller (PLC)* pada dasarnya adalah sistem komputer yang didesain untuk kepentingan khusus. *PLC* memiliki *input* dan *output* yang mempunyai fungsi mirip

dengan komputer. Kemiripan PLC dengan komputer dapat dilihat dari kemiripan struktur yang membentuknya. *Programmable Logic Controller* (PLC) memiliki tiga komponen utama yaitu; *processing*, *memory*, *input / output* (I/O). Sistem PLC di gambarkan pada gambar 2.3 sebagai berikut



Gambar 2.4. Blok Diagram Arsitektur PLC

IV.3.4. *Central Processing Unit (CPU)*

Central Processing Unit (CPU) sering hanya disebut *Processor* saja terdiri dari 3 utama :

1. *Register*
2. *Automatic Logic Unit* (ALU)
3. *Control Unit*



Gambar 2.5. Blok Diagram CPU

1. Register

Register merupakan bagian dari memori yang berkecepatan tinggi digunakan untuk menyimpan informasi yang penting selama *CPU* beroperasi. Macam-macam *register* adalah sebagai berikut :

➤ *Accumulator* (Ac)

Adalah tempat mengolah data, misal untuk membawa atau mengisi memori menambah, mengurangi, dan membandingkan. Bila data yang diolah lebih dari sebuah, $A + B$ misalnya, salah satunya harus berada di Ac kemudian hasil olah disimpan di Ac pula. Jadi isi Ac berubah setelah olah data. Terolah kedua *(operand)* berada di *register* lain atau memori terolah ini tidak berubah nilai setelah olah data.

➤ *Instruction register* / Register instruksi (IR)

Mempunyai dua tugas :

1. sebagai register serbaguna
2. sebagai penunjuk indeks (nomor urut)

Kegunaan mengindeks berhubungan dengan tabel atau daftar, misalnya menyalakan peraga 7 segmen (*seven segment*). Instruksi-instruksi suatu program terdiri atas kode operasi (*opcode*) dan medan alamat (*address field*). IR menguraikan bagian *opcode* nya saja suatu instruksi yang diterima dari "*register buffer data*" selanjutnya CU akan men"*decode*" isi IR dan membangkitkan sinyal-sinyal kendali tertentu sesuai dengan instruksi yang di"*decode*" tersebut.

➤ Program counter (PC)

Adalah tempat catatan langkah (alamat) program. Isi program counter bukan letak (alamat) sekarang (yang sedang dikerjakan) tetapi menunjuk alamat intruksi berikutnya (isi $PC = PC + 1$).

➤ *Stack Pointer* (SP)

Stack Pointer (SP) ini berisi alamat *stack* yang siap dipakai. Bila CPU melakukan tugas yang menyangkut *stack* maka ia juga mengubah isi SP agar selalu menunjukkan alamat *stack* yang siap dipakai. *Stack* sendiri adalah bagian dari RAM yang dicadangkan untuk mencatat isi register. *Stack Pointer* (SP) adalah register yang berisi alamat sedangkan *stack* adalah RAM. Operasi *stack* mengikuti kaidah FILO (*first in Last out*) atau LIFO (*last in first out*), kaidah tersebut berarti data yang pertama kali disimpan akan dikeluarkan paling akhir atau data yang paling akhir disimpan akan dikeluarkan paling awal.

➤ *Condition Code Register* (CC)

Berisi indikator (*flag*) yang menandai hasil perintah terakhir (yang baru saja dikerjakan). Rambu yang umum adalah :

- H = *half carry*, tanda bahwa hasil olah lebih dari setengah jatah bit
- C = *carry / borrow*, tanda bahwa hasil olah lebih dari jatah bit
- Z = *zero*, tanda bahwa hasil olah adalah nol
- N = *negative*, tanda bahwa hasil olah adalah negatif
- I = *interrupt mask*, tanda apakah sela akan dilayani atau tidak.

2. ALU (*Arithmetic Logic Unit*)

ALU membentuk operasi hitungan dan nalar terhadap *operand-operand* yang mapan sementara dalam register-register setelah operasi ALU juga menempatkan hasilnya dalam register. Biasanya hasil ALU ditempatkan dalam register khusus yang disebut *accumulator* (Ac).

3. *Control unit* (CU)

CU adalah suatu unit yang berfungsi untuk membangkitkan sinyal-sinyal cara sekuensial yang pulsanya diperoleh dari *clock* (detak).

CU membangkitkan dua buah kelompok sinyal yaitu :

- a. Sinyal-sinyal kendali internal untuk mengaktifkan ALU dan membuka / menutup lintasan data antara register-register.
- b. Sinyal-sinyal kendali eksternal ditujukan untuk memori dan I/O. Sinyal-sinyal tersebut dikirimkan untuk mengaktifkan operasi transfer data atau menanggapi interupsi dan permintaan bus.

IV.3.5. Memori

Yang di maksud dengan memori adalah merupakan sub sistem dari sistem mikroprosesor yang berfungsi untuk menyimpan program atau data. Penyimpanan data ada dua macam yaitu :

- a. ROM (*Read only Memory*)
- b. RAM (*Random Access Memory*)

ROM merupakan piranti penyimpan operating sistem untuk PLC, sedangkan RAM digunakan untuk menyimpan sementara proses perhitungan dan penalaran. Oleh karena PLC mempunyai ukuran yang sangat kecil maka karakteristik dari memory yang perlu diperhatikan adalah meliputi :

- Kecepatan
- Kepadatan
- Antisipasi Daya
- Harga komponen
- Faktor-faktor lain

ROM (*Read Only Memory*)

Kalau dalam komputer ROM adalah tempat untuk menyimpan program dasar untuk input dan output (BIOS), maka pada PLC ROM. digunakan untuk menyimpan sistem operasi untuk PLC itu sendiri.

RAM (*Random Access Memory*)

RAM adalah tempat untuk menyimpan data secara tidak permanen (*volatile*). Artinya RAM akan hilang bila tidak ada catu daya. RAM dikatakan random acak karena cara akses secara random (tidak menurut aturan-aturan tertentu, misal sekuensial). RAM sering disebut *Read / Write Memory* karena memori ini berfungsi untuk operasi baca (transfer data dari RAM ke CPU) dan operasi tulis (transfer data dari CPU ke RAM). RAM ada dua jenis yaitu SRAM (*Static RAM*), jenis memory dengan gel-gel memori tersusun atas beberapa flip-flop (Bistabil flip-flop). Jenis lain yaitu DRAM (*Dynamic RAM*).

Sel-sel memori tersusun atas kapasitor-kapasitor, sehingga memerlukan sinyal penyegaran (*fresh*). Pada PLC jenis memory yang digunakan jenis DRAM yang digunakan untuk mendukung CPU, jenis ini sudah memenuhi syarat.

Perbedaan antara *Static RAM* dan *Dynamic RAM* adalah sebagai berikut :

Static RAM

- Kecepatan Tinggi
- Kepadatan Rendah
- Perkawatannya sederhana
- Tidak memerlukan sinyal penyegar (*refresh*)

Dynamic RAM

- Kecepatan Rendah
- Kepadatan Tinggi
- Kebutuhan Daya Relatif Rendah
- Memerlukan Sinyal Penyegar.

PLC dilengkapi pula dengan EEPROM yang digunakan untuk menyimpan data program yang dikirim ke PLC, sehingga pada kondisi RUNNING keberadaan komputer atau console program tidak diperlukan lagi; atau pada saat terjadi gangguan pada catu daya maka program akan tetap tersimpan dalam PLC.

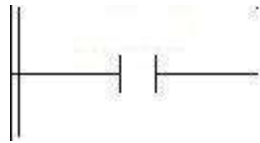
IV.4. SISTEM PLC

IV.4.1. Struktur Dasar PLC

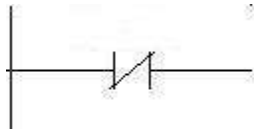
1. *Central Prosesing Unit(CPU)*
2. *Memory*
3. *Input / Output*
4. *Power Supply*

IV.4.2. Simbol – simbol *ladder diagram*

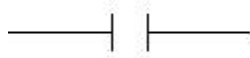
1. **Load / LD** = *Star* pada *normally open input*



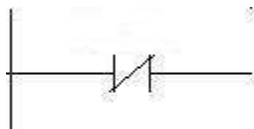
2. **Load Not / LD NOT** = *Star* pada *normally close input*



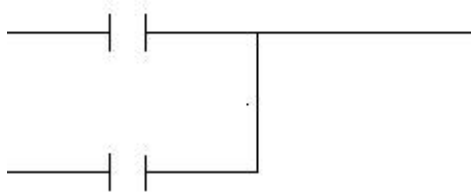
3. **AND** = menghubungkan dua atau lebih *input* dalam bentuk *normally open* secara seri.



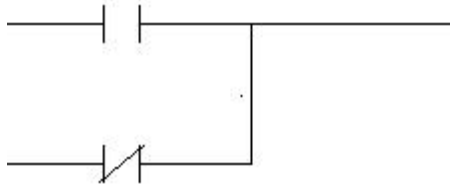
4. **AND NOT** = menghubungkan 2 atau lebih *input* dalam bentuk *normally close*.



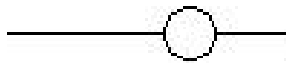
5. **OR** = menghubungkan 2 atau lebih *input* dalam bentuk *normally open* secara *paralel*.



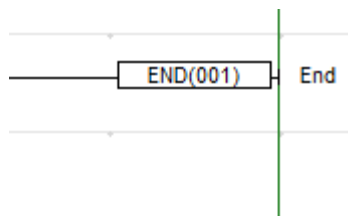
6. **OR NOT** = menghubungkan 2 atau lebih *input* dalam bentuk *normally close* secara *paralel*.



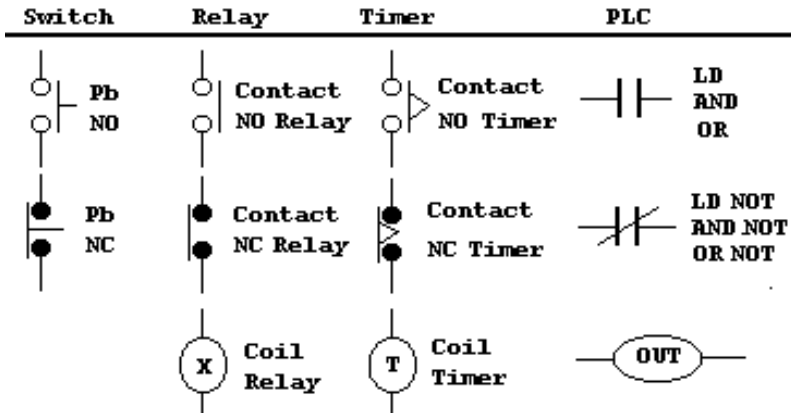
7. **OUTPUT / OUT** = menyalakan *output*.



8. **END** = mengakhiri program

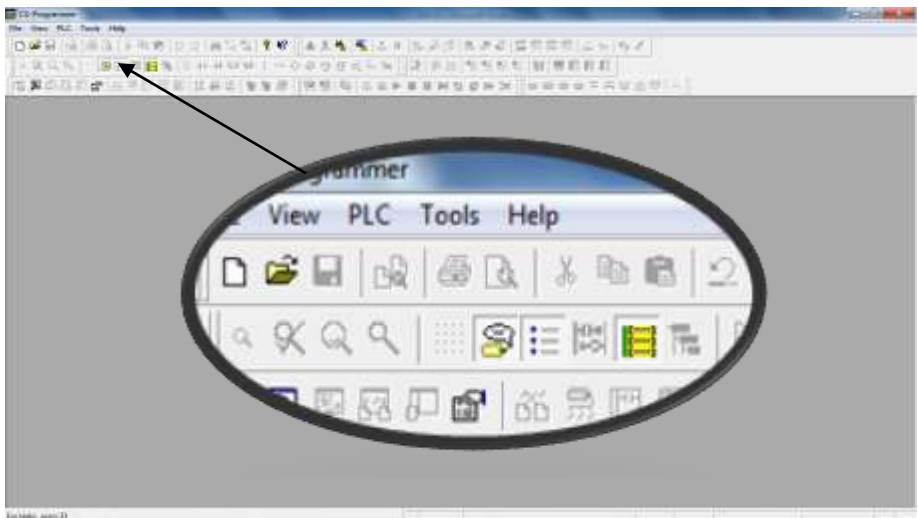


IV.4.3. Konversi Symbol

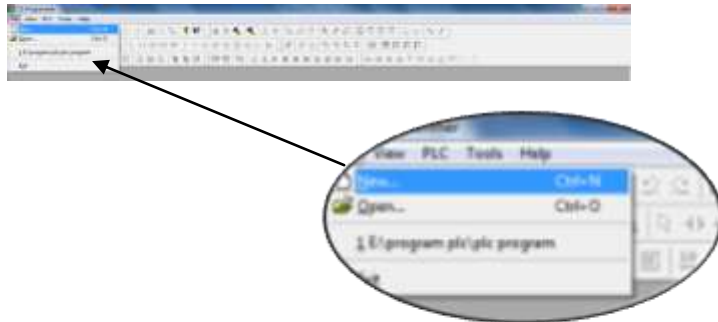


IV.4.4. Membuat programan PLC

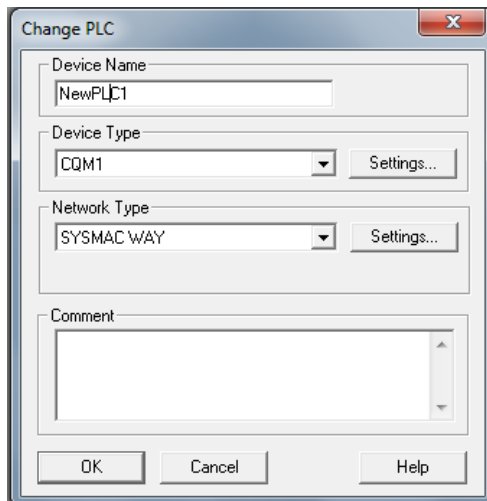
Pemograman PLC CQM1 CPU21 menggunakan LSS (*Ladder Support Software*) *CX-One Programmer*. Untuk memulai membuat program pada *cx programmer* ada beberapa hal yang harus diketahui. Pertama buka aplikasi *cx programmer* dan akan muncul jendela seperti pada gambar dibawah ini. Setelah itu klik File dan pilih new untuk membuat program baru.



Dengan menggunakan *pointer mouse*, tekan *File- New* untuk membuat program

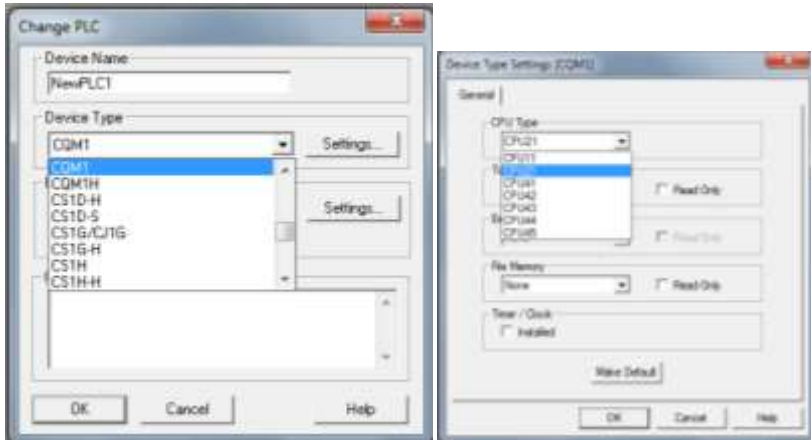


Setting PLC dan koneksi PLC dibuat setelah proses pembuatan program baru. Akan muncul jendela sebagai berikut :

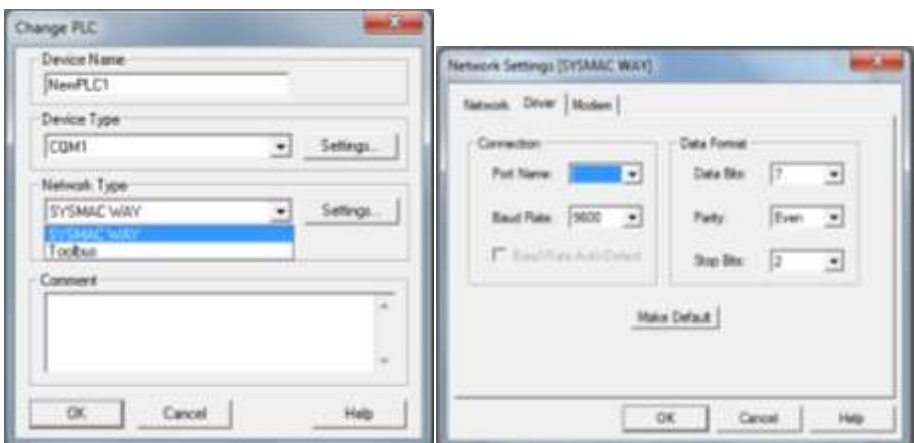


Pada Change PLC, Device Name nama default adalah NewPLC1 dapat diubah dengan nama apapun sesuai dengan keinginan pembuat program. Setelah itu dilanjutkan dengan pemilihan PLC yang sesuai dengan yang digunakan. Dalam hal ini digunakan seri CPM1. Pada kotak setting, pilih CPU

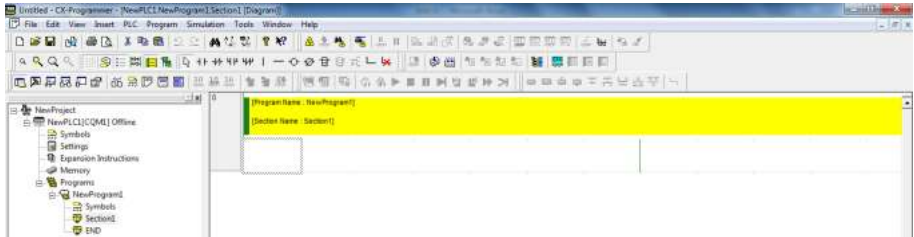
PLC CPM1 yang sesuai, dapat dilihat pada bodi PLC. PLC yang digunakan di sini menggunakan CPU 21. Setelah proses pemilihan dilakukan, tekan ok



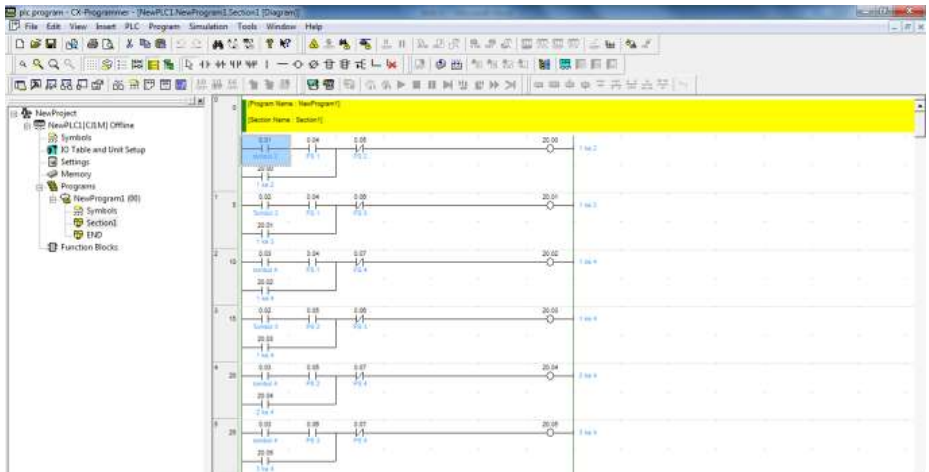
Pada bagian change PLC, sub-bagian Network Type, pilih YSMAC-WAY, kemudian pada kotak Seting lakukan pengaturan pada COM komputer yang digunakan



Setelah itu pilih, tekan ok. Akan muncul jendela pembuatan program pertama kali seperti gambar berikut. Setelah muncul jendela ini, maka komputer siap digunakan untuk pembuatan program

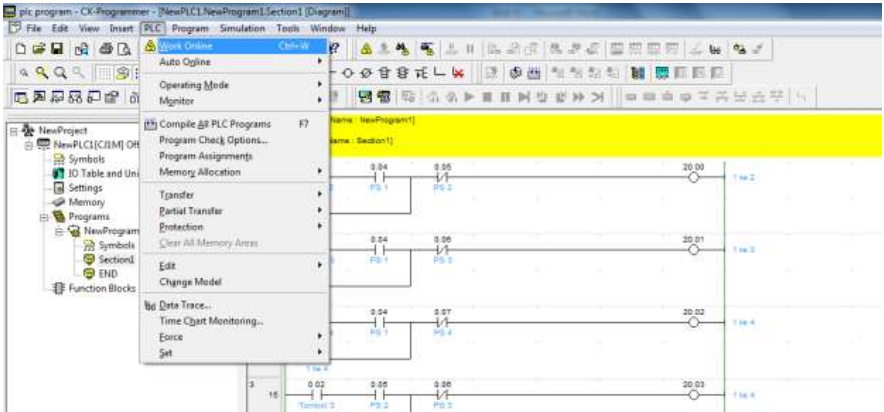


Pembuatan Program

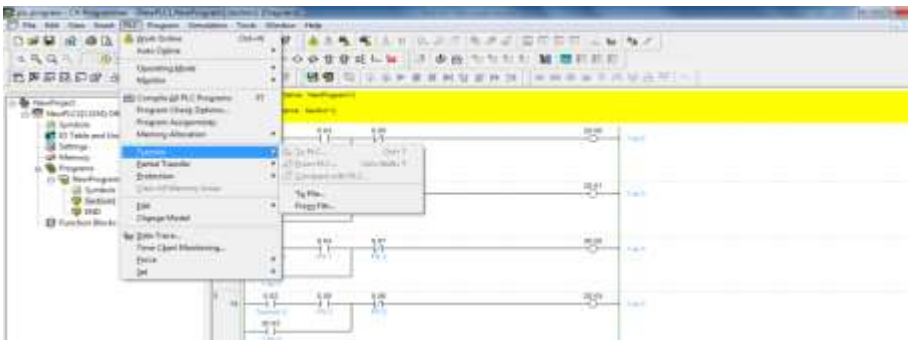


Men-transfer dan menjalankan program

Program yang selesai dibuat, harus di *transfer* dulu kedalam memori program PLC. Sebelum di*transfer*, PLC harus dihubungkan dulu dengan komputer. Cara menghubungkannya adalah dengan menekan PLC *Work Online*, akan muncul jendela berikut :



Jika koneksi sudah benar, maka program yang dibuat dapat di *transfer* ke dalam memori program PLC, caranya adalah tekan *PLC Transfer To PLC*



Saat proses transfer



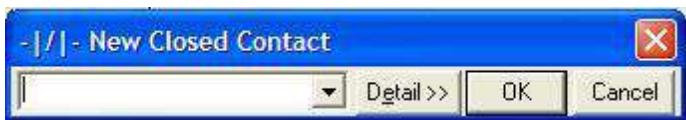
Cara Memasukkan Simbol Ladder Diagram

Setelah itu anda bisa mulai memasukkan symbol-symbol ladder. Diagram untuk membuat kontak N/O tekan huruf C maka akan muncul seperti gambar dibawah ini :



Setelah itu isikan address pada kolom misalnya address input 0.01 lalu klik detail untuk membuat nama/comment input tsb lalu klik OK.

Untuk memasukkan simbol N/C tekan / (garis miring) dan isi alamat input/output pada kolom. Lihat gambar dibawah ini :



setelah itu isi data pada kolom ,klik detail isi comment lalu klik OK.

Untuk membuat Output, tekan huruf O pada keyboard maka akan muncul seperti gambar di bawah ini :



setelah itu isi address output pada kolom misalnya 10.00 lalu klik detail isi comment dan selanjutnya klik OK.

Unutk membuat instruksi : Timer, Counter, Mov, Scaling, Difu, IL – ILC dll. Tekan hutuf I lalu akan muncul seperti gambar di bawah ini:



Ketik pada kolom, nama Instruksi yang akan dipakai misalnya ketik TIM untuk Timer, ketik CNT untuk Counter dll. Setelah itu klik detailnya. Selanjutnya untuk memasukkan symbol yang lain :

Ctrl + ® = untuk membuat garis horisontal.

Ctrl + ¯ = untuk membuat garis vertikal

Tekan huruf R untuk menambah network ditengah2 network yang lain.

ADDRESS / ALAMAT PADA PLC OMRONCPM1A

Input ch 0 yaitu dari 0.00 - 0.10 (sesuai type PLCnya)

Input ch 1 yaitu dari 1.00 - 1.12 (sesuai type PLCnya)

Output ch 10 yaitu dari 10.00 - 10.15 (sesuai type PLCnya)

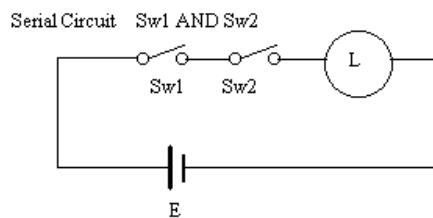
Output ch 11 yaitu dari 11.00 - 11.15 (sesuai type PLCnya)

Internal Relay yaitu dari 600 – 1500 (tergantung type PLC)

IV.4.5. Rangkaian Dasar Kontroller

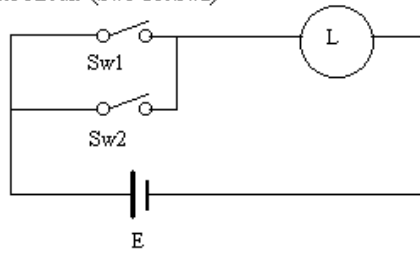
1. Rangkaian *Serial*
2. Rangkaian *Paralel*
3. Rangkaian *Self Hold*
4. ON Delay Timer
5. OFF Delay Timer

1. Rangkaian *Serial*



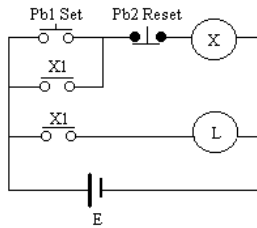
2. Rangkaian *Paralel*

Paralel Circuit (Sw1 OR Sw2)

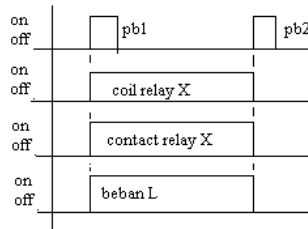


3. Rangkaian *Self Hold*

Self Hold Circuit

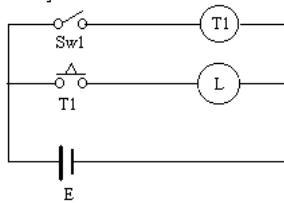


Timing Chart

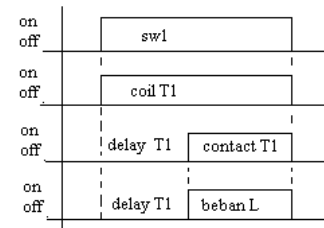


4. ON Delay Timer

ON Delay Timer Circuit

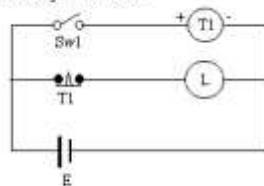


Timing Chart

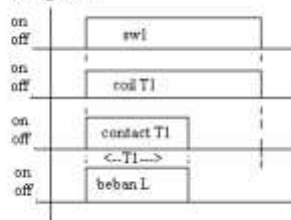


5. OFF Delay Timer

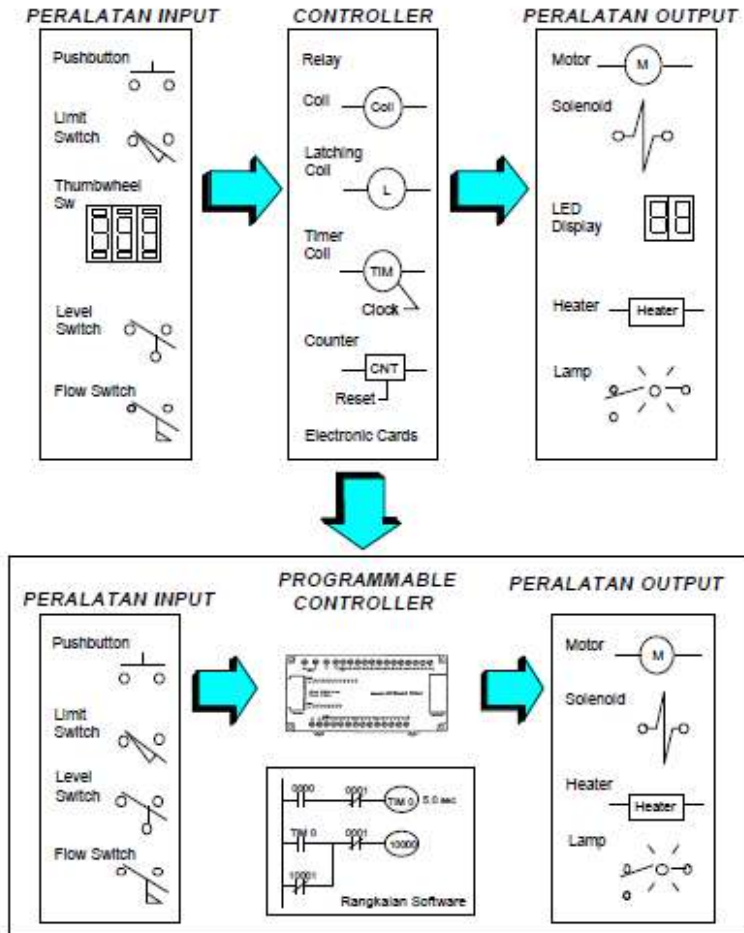
OFF Delay Timer Circuit



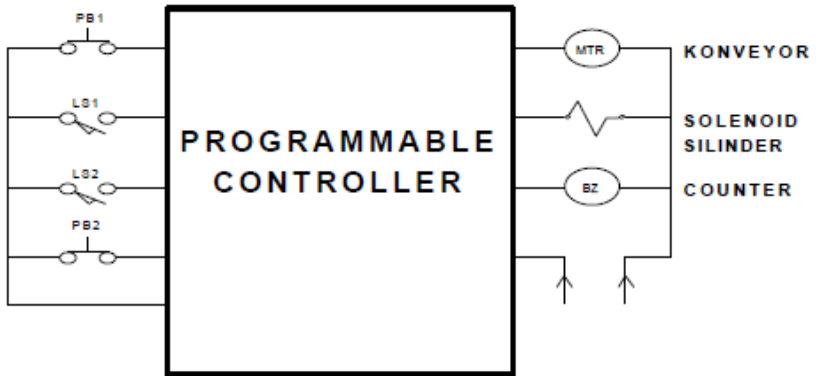
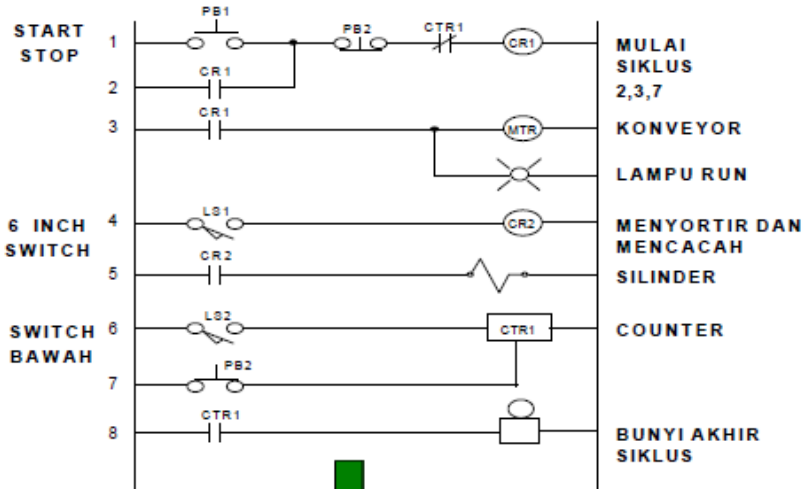
Timing Chart



Prinsip dasar *Controller*



Bandingkan kontor konvensional dengan PLC



IV.5. APLIKASI PLC

PLC banyak diterapkan pada

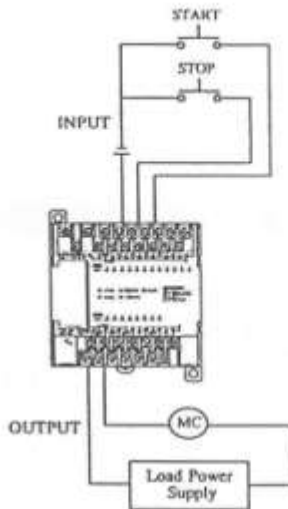
- Material Handling
- Conveyor System
- Packaging Machine
- Pick and Place Robot Control
- Pump Control
- Swimming Pool
- Paper and Pulp Industries
- Glass Manufacturing
- Precast Concrete Industries
- Cement Manufacturing
- Printing Industries
- Electro-plating Plants
- Food Processing
- Machine Tools
- Tobacco Industries
- Plastic Moulding Machine
- Semi-conductor Manufacturing Machine
- Sugar Manufacturing Plant
- Palm Oil Manufacturing Plant
- Air Condition Control
- Power Station Plant
- Process Monitoring Control
- Electrical/Electronic Appliance Manufacturing
- Disk Drive Manufacturing
- Petrol Chemical Plant
- Plastic Manufacturing Industries
- Car Manufacturing Plant
- Iron and Steel Mill
- Dairy Product Manufacturing Plant
- Building Automation
- Tyre Manufacturing
- Integrated Circuit Chip Manufacturing
- Sewage Treatment Plant
- Security Control System
- Lift Control System
- Generator Control System
- Amusement Park Control
- TV Manufacturing Plant

Berikut ini contoh penerapan PLC (diambil jenis SYSMAC CPM1) untuk beberapa penerapan.

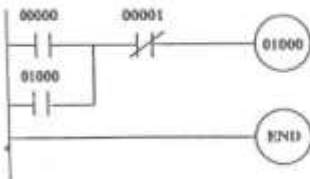
■ Application #1: Self Holding Circuit

Input	Device	Output	Device
00000	Start Pb.	01000	Motor
00001	Stop Pb.		

Circuit Diagram



Ladder Diagram



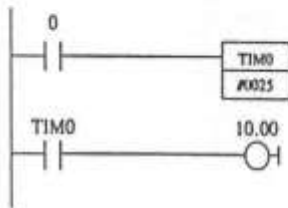
Mnemonic Codes

Address	Instruction	Device
0000	LD	00000
0001	OR	01000
0002	AND NOT	00001
0003	OUT	01000
0004	END(01)	

Note: This program enables the output 01000 to stay "ON" despite the status of the input 00000 which triggered it. This is often desirable in continuous output operations which begin with only an instantaneous ON input.

■ Application #2 : On Delay Circuit

Ladder diagram



Mnemonic codes

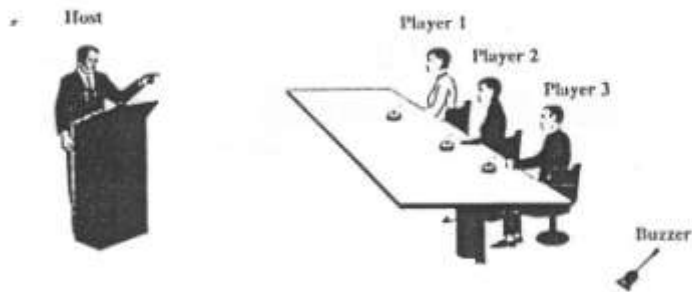
Address	Instruction	Data
00000	LD	00000
00001	TIM	000
		#0025
00002	LD	TIM 000
00003	OUT	01000
00004	END(01)	

Timing diagram



TIM is a decrementing ON-delay timer instruction which requires a timer number and a set value (SV) ranging from 0000 to 9999 (0 to 999.9 seconds).

■ Example : Priority Determination Design



The game buzzer control requirement:

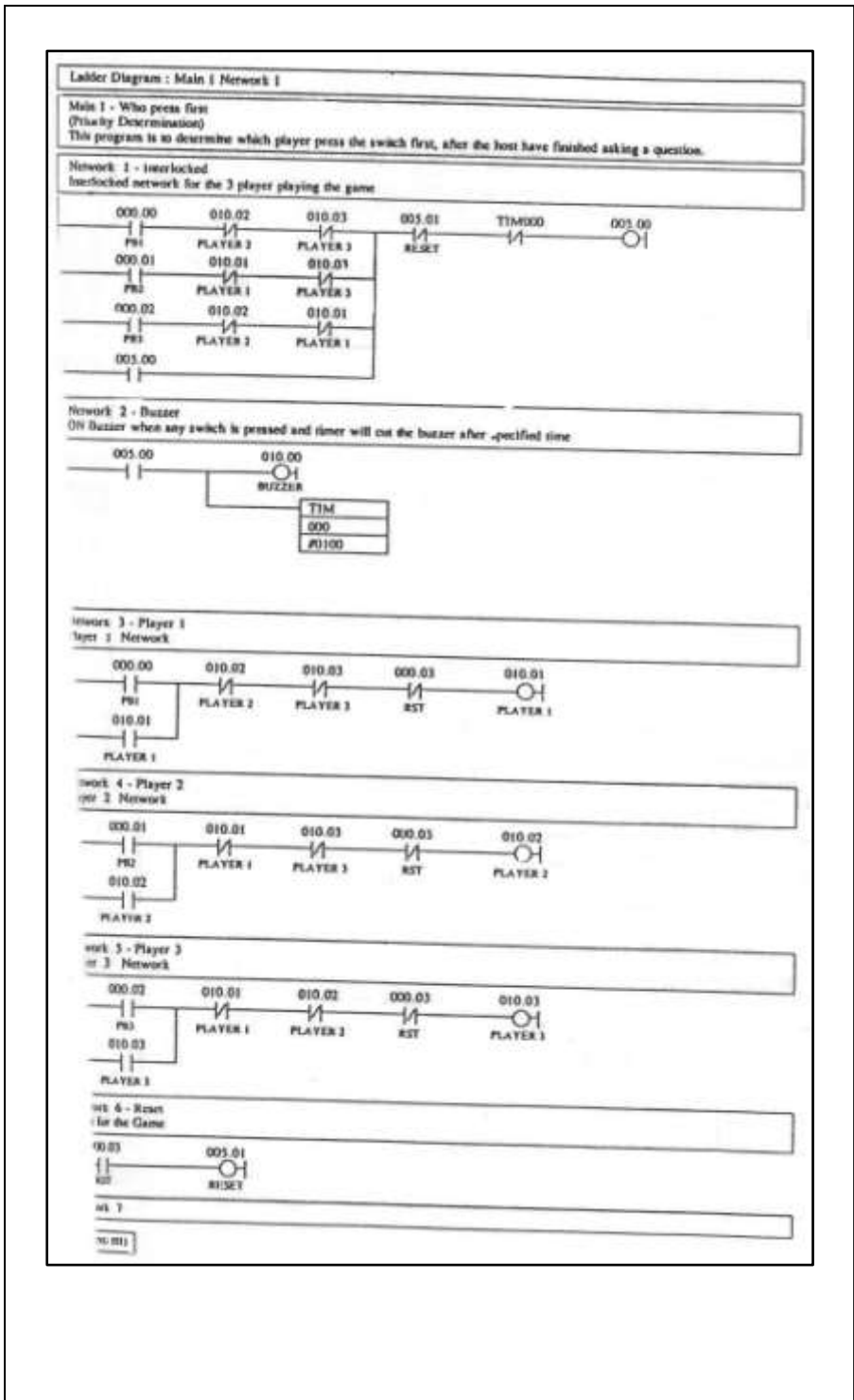
1. After the Host has finished with question.
2. The 3 players will press the switch in front of them to fight to be first to answer the question.
3. The buzzer will sound for 10 sec after any one of the players has touched the switch.
4. The light indicator in front of each player will light-up and will only be reset by the Host switch.

• I/O Assignment

Input	Device
00000	PB1
00001	PB2
00002	PB3
00003	RST (reset)

Output	Device
01000	Buzzer
01001	Player 1 light
01002	Player 2 light
01003	Player 3 light

By Using SYSWIN Programming Software, the design of the buzzer control ladder diagram is shown as follow.



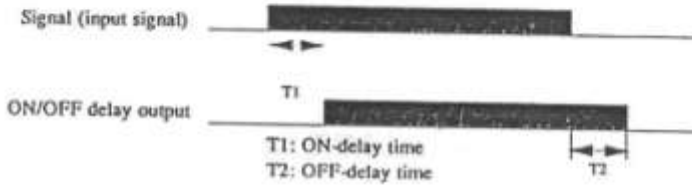
■ On and Off Delay Circuit

This circuit is used to delay the ON/OFF time of an input signal for a given time.

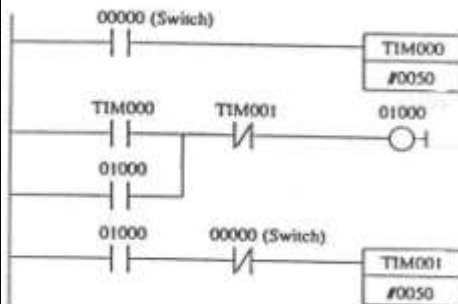
I/O Assignment

Input	Device	Output	Device
00000	Switch	01000	Motor

Timing diagram



Ladder diagram



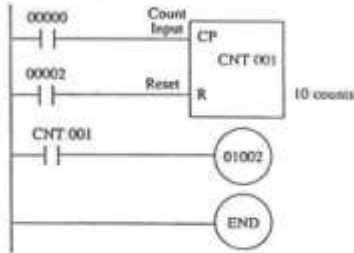
Mnemonic Codes

Address	Instruction	Device
00000	LD	00000
00001	TIM	000
		#0050
00002	LD	TIM000
00003	OR	01000
00004	AND-NOT	TIM0001
00005	OUT	01000
00006	LD	01000
00007	AND-NOT	00000
00008	TIM	001
		#0050
00009	END(01)	

■ Counter

CNT (counter) is a preset decremental counter. That is, it decrements one count everytime an input signal goes from OFF to ON. The counter must be programmed with a count input, a reset input, a counter number and a set value (SV). The Counter number can range from 000 to 511 and the Set value (SV) can range from 0000 to 9999.

Ladder diagram



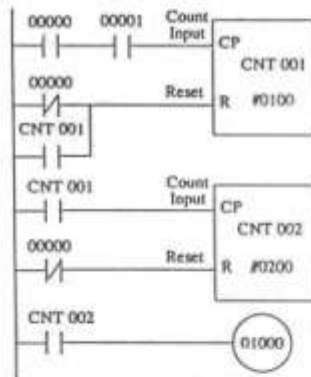
Mnemonic codes

Address	Instruction	Operands
0000	LD	00000
0001	LD	00002
0002	CNT	001
		#0010
0003	LD	CNT 001
0004	OUT	01002
0005	END(01)	

■ Use of CNT and TIM instructions

- Example #1: Count to 20,000 counts

Ladder diagram

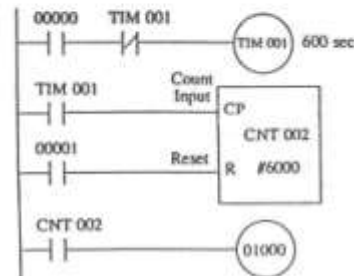


Mnemonic codes

Address	Instruction	Operands
00000	LD	00000
00001	AND	00001
00002	LD-NOT	TIM 000
00003	OR	CNT 001
00004	CNT	001
		#0100
00005	LD	CNT 001
00006	LD-NOT	00000
00007	CNT	002
		#0200
00008	LD	CNT 002
00009	OUT	01000
00010	END(01)	

- Example #2: Prolong time range to 1,000 hours

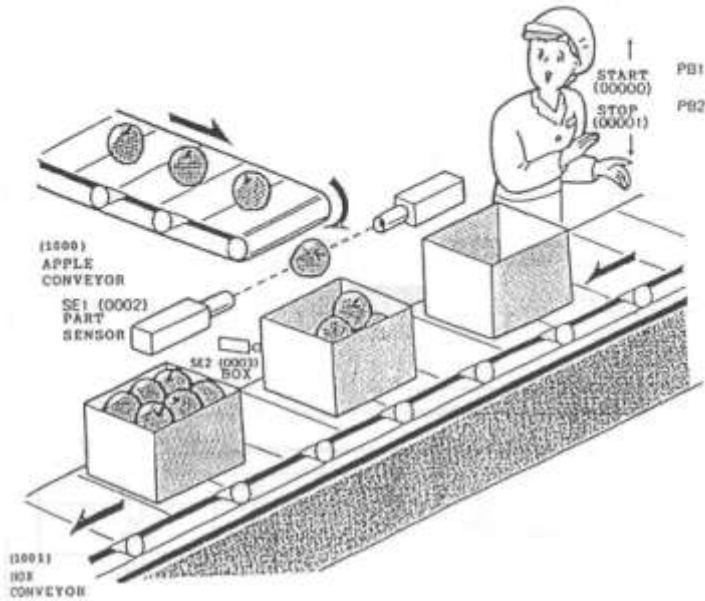
Ladder diagram



Mnemonic codes

Address	Instruction	Operands
00000	LD	00000
00001	AND-NOT	TIM 001
00002	TIM	001
		#6000
00003	LD	TIM 001
00004	LD	00001
00005	CNT	002
		#6000
00006	LD	CNT 002
00007	OUT	01000

■ Example : Packaging Line Control

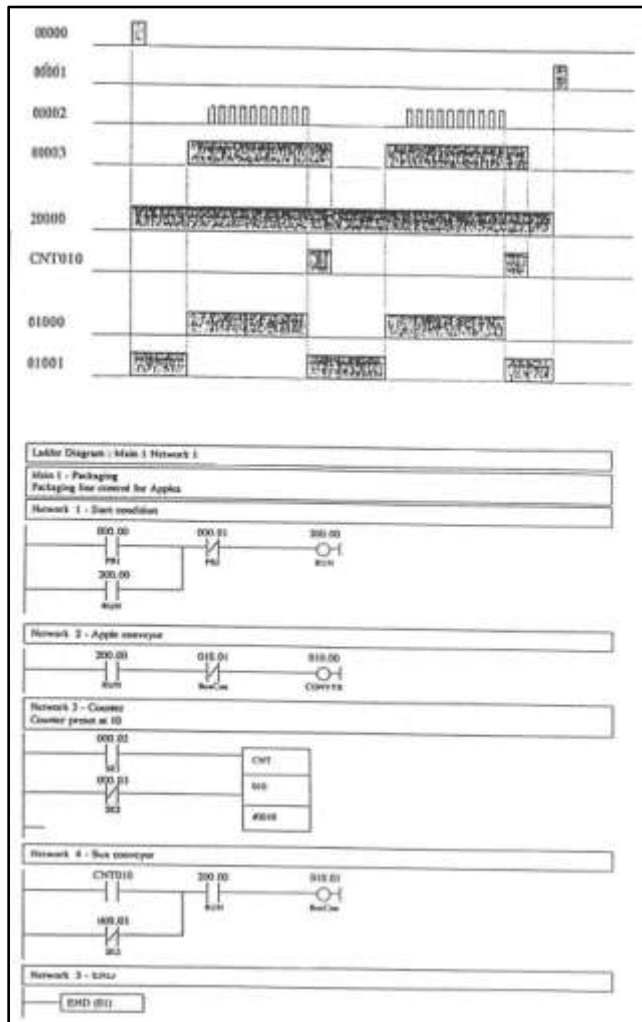


■ Operation

When PB1 (START Push Button) is pressed, the box conveyor moves. Upon detection of box present, the box conveyor stops and the Apple conveyor starts. Part sensor will count for 10 apples. Apple conveyor stops and box conveyor starts again. Counter will be reset and operation repeats until PB2 (STOP Push Button) is pressed.

Input	Devices
00000	START Push Button (PB1)
00001	STOP Push Button (PB2)
00002	Part Present (SE1)
00003	Box Present (SE2)

Output	Devices
01000	Apple Conveyor
01001	Box Conveyor

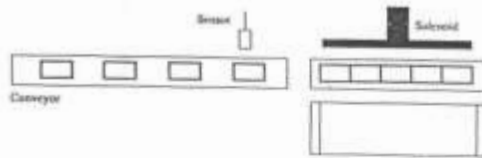


Mnemonic Codes

Address	Instruction	Oper	Address	Instruction	Oper
0000	LD	00000	0008	LD NOT	00003
0001	OR	20000	0009	CNT	010
0002	AND NOT	00001			#0010
0003	OUT	20000	0010	LD CNT	010
0004	LD	20000	0011	OR NOT	00003
0005	AND NOT	01001	0012	AND	20000
0006	OUT	01000	0013	OUT	01001
0007	LD	00002	0014	END (01)	

Application #3: Control Circuit For Packaging Machine

The control circuit is used to detect and count the number of products being carried on an assembly line. When it counts five products, the circuit energizes a solenoid. The solenoid is energized for a period of two seconds and is then shut off, causing it to retract.

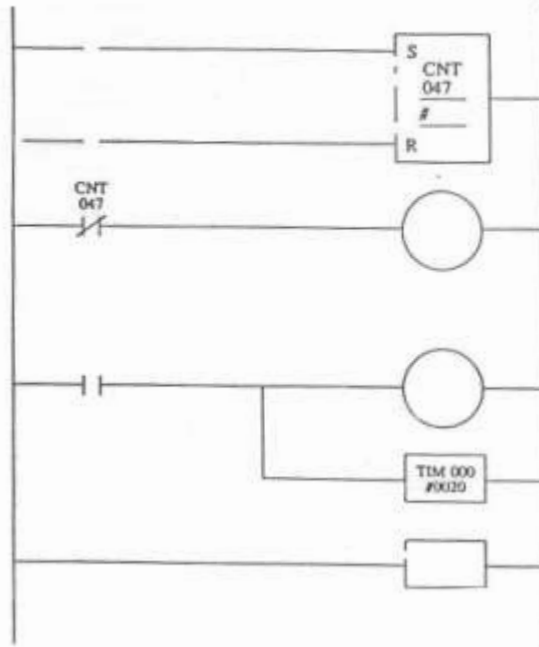


Input	Output
00000	Sensor

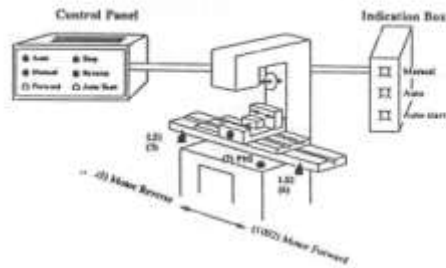
PLC Output	Device
01000	Conveyor
01001	Solenoid

Other	
CNT 047	Product counter
TIM 000	solenoid energize timer

Question: Fill in the gaps below to make this circuit operate.



Application #4: Drilling Control Operation



I/O Assignments

Input	Device
0000	Auto Switch
0001	Manual
0002	Forward Switch (SW1)
0003	Stop Switch (SW2)
0004	Reverse Switch (SW3)
0005	Limit Switch (LS1)
0006	Limit Switch (LS2)
0007	Auto Start Button (PB5)

Output	Device
1000	Auto Indicator
1001	Manual Indicator
1002	Motor Forward
1003	Motor Reverse
1004	Auto Start Indicator

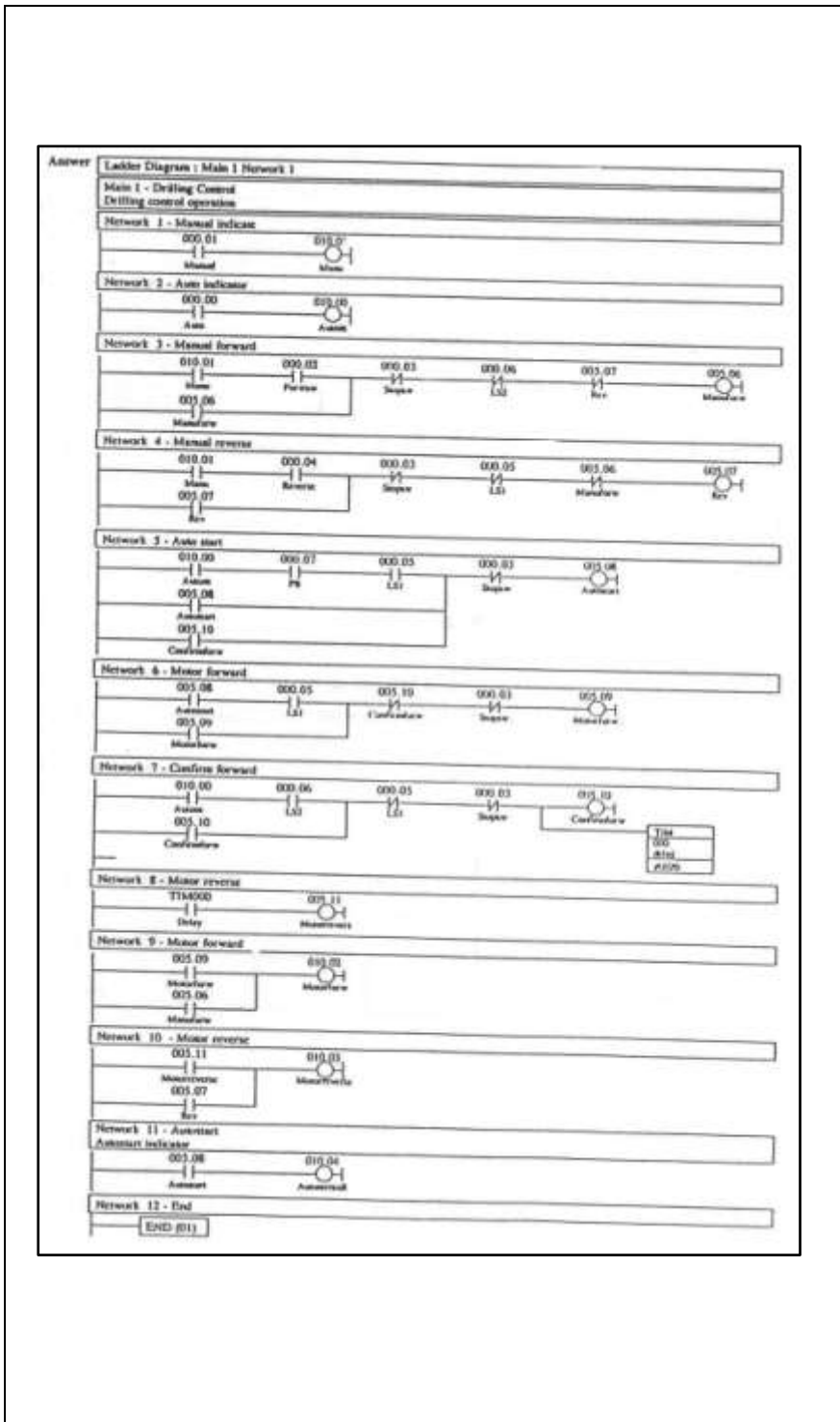
Procedure

1. Manual Operation

- 1.1 When SW1 is ON, Motor moves forward. It can be stopped by SW2. When the drill touches LS2, the Motor is cut-off.
- 1.2 When SW3 is ON, Motor moves in reverse. It can be stopped by SW2. When the drill touches LS1, the Motor is cut-off.

2. Auto-cycle

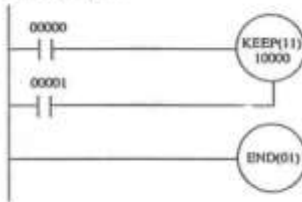
- 2.1 When PB and LS1 is ON, the Motor moves forward until LS2 is activated. The Timer then starts timing down. The Motor reverses when the timer reaches 2 seconds. When it returns to LS1 position, the cycle is repeated.



■ KEEP(11) - Latching relay

KEEP is used as a latch. It maintains an ON or OFF state of a bit until one of its two inputs sets or resets it. If the KEEP function is used together with a IIR relay the state of the latched output is retained even during a power failure.

Ladder diagram



Mnemonic codes

Address	Instruction	Comment
0000	LD	00000
0001	LD	00001
0002	KEEP(11)	10000
0003	END(01)	

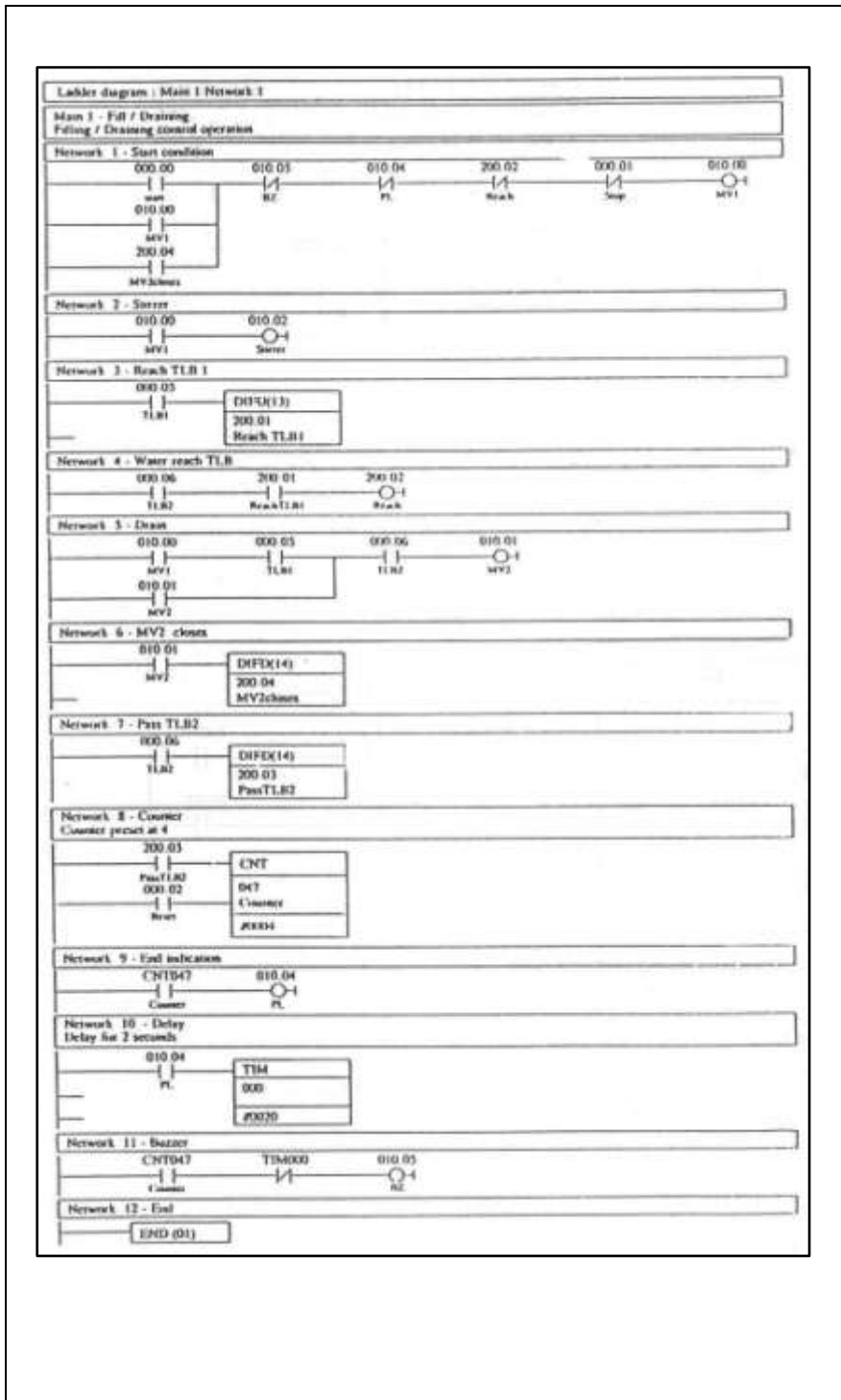
■ DIFU(13) and DIFD(14) - Differentiation

DIFU and DIFD turns an output ON for one scan only.

DIFU turns its output ON when it detects an OFF -> ON transition in its input signal.

DIFD turns its output ON when it detects an ON -> OFF transition in its input signal.

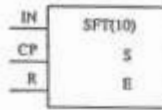




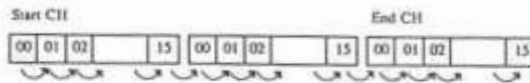
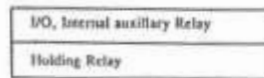
■ Shift Register - SFT(10)

Shift Register (SFT) shifts a 16-bit data in specified channels by 1 bit. Although this instruction shifts data within channels, both a start channel and an end channel must be specified as the data.

Ladder Symbols

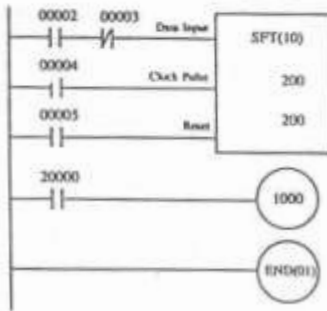


Operand Data Areas



Example circuit

Ladder diagram



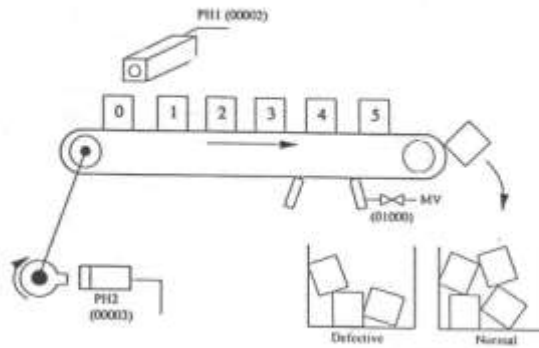
Mnemonic code

Address	Instruction	Area
0000	LD	00002
0001	AND NOT	00003
0002	LD	00004
0003	LD	00005
0004	SFT(10)	
		200
		200
0005	LD	20000
0006	OUT	1000
0007	END(01)	

- **NOTE :** When a reset input is applied to the Shift Register, all 16 bits are reset together. If the Holding Relay area is used, the data are retained during power failure.

■ Application #7: Parts Sorting

In this application, defective products are detected and rejected from those being carried on the conveyor.

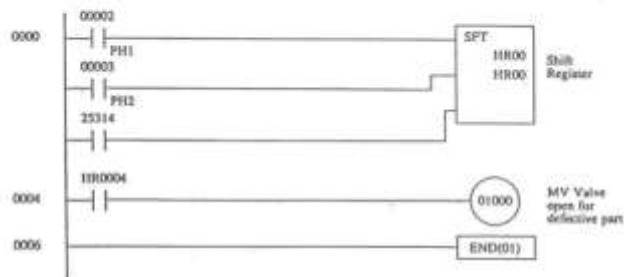


Photoelectric sensor (PH1) serves as the data input to the shift register. The signal output from this sensor turns ON when a defective product has been detected; otherwise it remains OFF.

Photoelectric sensor (PH2) is used as a clock generator that serves as the clock input to the shift register. It generates one pulse each time the product, spaced at a fixed interval from each other, has travelled a predetermined distance.

From the moment a defective product is detected by PH1, it is traced by the shift register until the product arrives at the predetermined position on the conveyor where it is ejected by the magnetic valve MV.

Ladder diagram



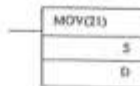
Mnemonic codes

Address	Mnemonic	Value
0000	LD	00002
0001	LD	00003
0002	LD	25314
0003	SFT	HR00
		HR00
0004	LD	HR0004
0005	OUT	01000
0006	END(01)	

■ Move - MOV(21)

MOV transfer source data (either the data in a specified channel or a four digit hexadecimal constant) to a destination channel. Therefore, MOV requires two data parameters to be specified: the source channel or constant and the destination channel.

Ladder Symbols



Operand Data Areas

S: Source channel	
IR, SR, AR, DM, HR, TC, LR, #	
D: Destination channel	
IR, AR, DM, HR, LR	

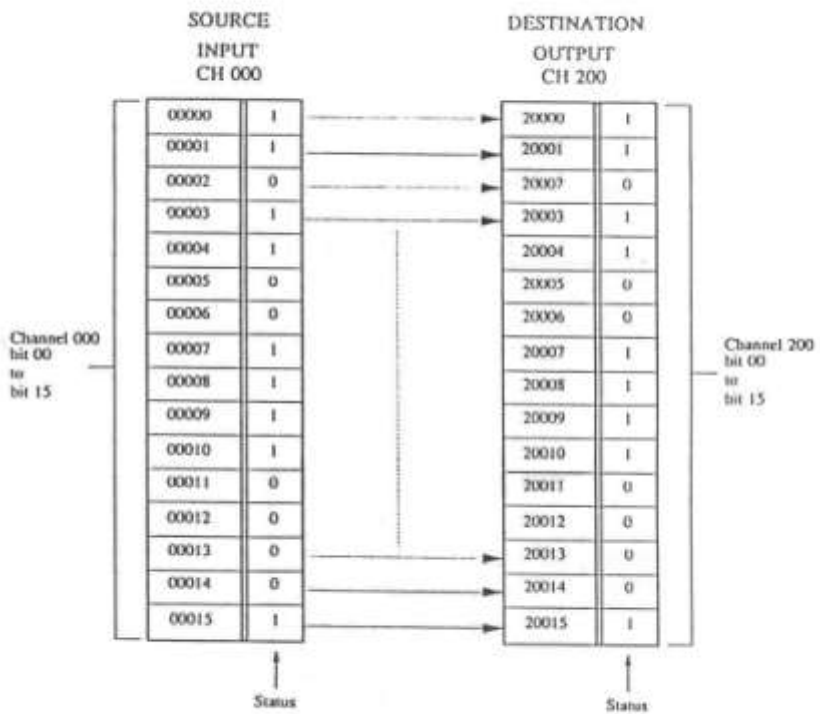
Example circuit

Ladder diagram



Address	Mnemonic	Value
0000	LD	25313
0001	MOV(21)	
		400
		200
0003	END(01)	

The following diagram illustrates the MOV operation:

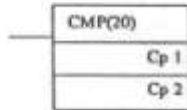


In this case, data in Input Channel 000 is moved to Output channel 200.

pare - CMP(20)

pare (CMP) is used to compare the data in a specific channel, with the data other channel, or a four-digit, hexadecimal constant. Therefore, two data be specified immediately after the CMP(20) instruction.

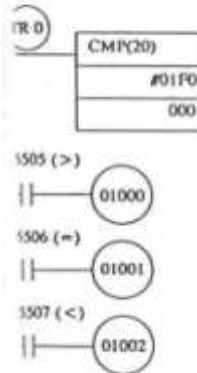
Ladder Symbols



Operand Data Areas

Cp 1: First compare word	
IR, SR, AR, DM, HR, TC, LR, #	
Cp 2: Second compare word	
IR, SR, AR, DM, HR, TC, LR, #	

ircuit ram



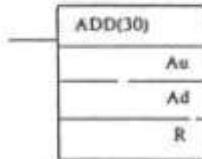
Address	Instruction	Operand
0000	LD	25313
0001	OUT	TR 0
0002	CMP(20)	
		#01F0
		000
0003	AND	25505
0004	OUT	01000
0005	LD	TR 0
0006	AND	25506
0007	OUT	01001
0008	LD	TR 0
0009	AND	25507
0010	OUT	01002

■ Add - ADD(30)

ADD totals the data in two different channels, or one channel and a constant and then outputs the sum to a third channel.

Therefore, three data parameters must be specified : an augend, an addend and a result channel.

Ladder Symbols



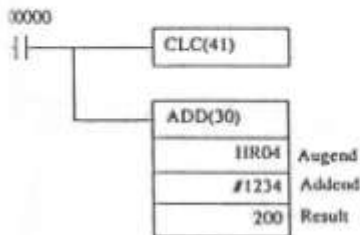
Operand Data Areas

Au: Augend channel
IR, SR, AR, DM, HR, TC, LR, #

Ad: Addend channel
IR, SR, AR, DM, HR, TC, LR, #

R: Result channel
IR, AR, DM, HR, LR

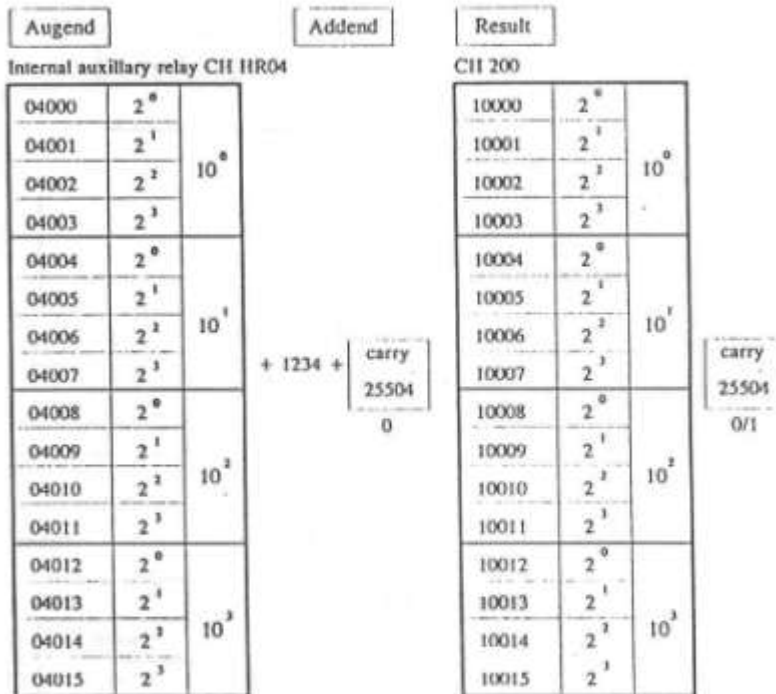
Example circuit Ladder diagram



Address	Instruction	Data
0000	LD	00000
0001	CLC(41)	
0002	ADD(30)	
		11R04
		#1234
		200
0004	END(01)	

In the program, when input 00000 is turned ON, the data in internal relay IIR040 is added to the constant 1234. The result of the addition is output to CH 200. If a carry is generated as a result of the addition, the carry flag (special relay 25504) is turned ON. If the result of the addition is 0000, special relay 25506 (the "Z" flag) is turned ON.

The following diagram illustrates the ADD operation.



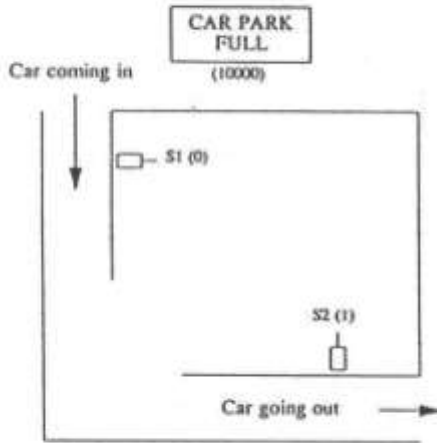
In the above example, before executing ADD, the Carry Flag/CY (special relay 25504) is turned OFF by the Clear Carry (CLC). The addition and subtraction instructions include CY in the calculation as well as in the result. Be sure to clear CY if its previous status is not required in the calculation, and to use the result placed in CY, if required, before it is changed by execution of any other instruction.

The augend and addend must be in BCD, if not special relay 25503 (Error Flag) is turned ON and ADD is not executed.

ADD is executed each time the CPU scans the program. To execute it only once.

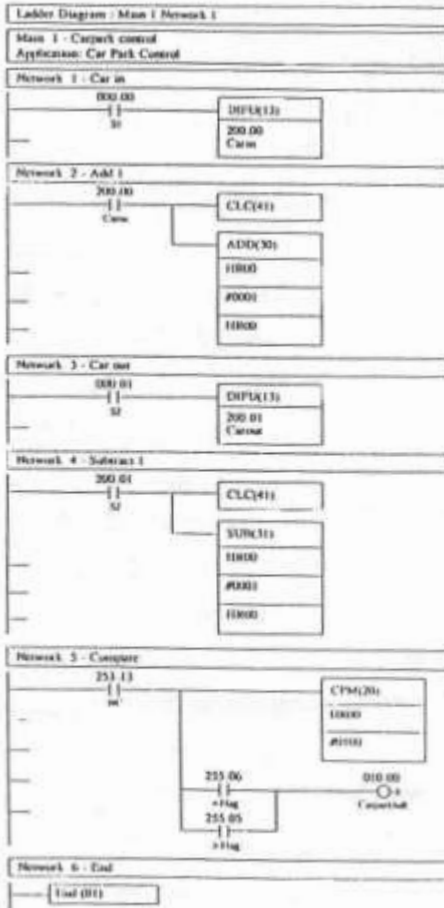
Application #15: Car Park Control

This is a simple car park control system that allows only a maximum of 100 cars parking space. Everytime a car comes in, the PLC will automatically add one through sensor S1. Every car that goes out will automatically be subtracted by one through sensor S2. When 0 cars are registered, the car park full sign will be lighted to inform oncoming vehicles not to enter.



I/O Assignment

Input	Device	Output	Device
00000	Sensor S1	01000	Car Park Full Sign
00001	Sensor S2		



Mnemonic codes

Address	Instruction	Value
0000	LD	00000
0001	DIFU(13)	20000
0002	LD	20000
0003	CLC(41)	
0004	ADD(30)	
		11R00
		#0001
		11R00
0005	LD	00001
0006	DIFU(13)	20001
0007	LD	20001
0008	CLC(41)	
0009	SUB(31)	
		11R00
		#0001
		11R00
0010	LD	25313
0011	OUT	TR 0
0012	CMP(20)	
		11R00
		#0100
0013	AND	25506
0014	LD	TR 0
0015	AND	25505
0016	OR LD	
0017	OUT	01000
0018	END(01)	

DAFTAR PUSTAKA

1. Joji P , 2008, Pneumatics Control, Wiley India Pvt. Limited,
2. H. Meixner, E. Sauer, Introduction to Electropneumatic, Festo Didactic KG
3. Herbert E. 1991, Merritt, Hydraulic Control System, Wiley
4. Peter Rohne, Automation with Programmable Logic Controllers, New South Wales Press, Sydney Australia