

MATERI PENGABDIAN PADA MASYARAKAT



Workshop Aplikasi PLC Di Industri Pada Jurusan Pendidikan Teknik
Elektro Fakultas Teknik Universitas Negeri Yogyakarta

Tanggal 7 – 16 November 2002

Oleh: Drs. Sukir, M.T.
NIP. 19621125 198702 1 001

Jurusan PT Elektro Fakultas Teknik
Universitas Negeri Yogyakarta
2002

Workshop Aplikasi PLC Di Industri Pada Jurusan Pendidikan Teknik Elektro Fakultas Teknik Universitas Negeri Yogyakarta

A. Pendahuluan.

Saat ini kemajuan ilmu pengetahuan dan teknologi yang diterapkan di berbagai bidang meningkat begitu pesat. Salah satu contoh kemajuan ilmu pengetahuan dan teknologi pada bidang ketenagalistrikan yang diterapkan di dunia industri dan dunia usaha adalah *Programmable Logic Controllers* (PLC). Menurut Festo (2004), PLC merupakan piranti elektronika digital dengan menggunakan memori yang dapat diprogram sebagai penyimpan internal dari sekumpulan instruksi, yang mengimplementasikan fungsi-fungsi tertentu seperti logika, sekuensial (*sequential*), pewaktuan (*timer*), perhitungan (*calculation*), dan aritmatika. PLC digunakan di dunia industri dan dunia usaha dari waktu ke waktu semakin maju dan canggih sebagai otak pengendali berbagai peralatan atau mesin. Sebagai contoh penggunaan PLC di dunia industri dan dunia usaha yaitu penggunaan PLC sebagai pengendali lif pada gedung bertingkat, pengendali ban berjalan (*conveyor belt*) untuk pengisian, penutupan dan pengepakan botol minuman, pengendali ban berjalan di bandar udara, pengendali pada pengelasan dan pengepresan bodi mobil, pengendali penyortiran produk industri, dan sebagainya.

Kecanggihannya PLC dalam mengendalikan peralatan dan mesin di industri ditunjukkan oleh kinerjanya yang cepat, cermat, akurat, tidak melakukan kesalahan, handal, dan kualitas pekerjaan yang baik. PLC yang memiliki keunggulan kinerja seperti tersebut di atas sangat menguntungkan pihak industri karena produktivitas dapat meningkat, efisiensi semakin tinggi, kualitas produk baik dan terjaga, dan sebagainya. Dengan semakin berkembangnya variasi produk PLC, maka saat ini PLC tidak saja hanya digunakan untuk pengendalian peralatan atau mesin canggih di industri saja, namun sudah berkembang aplikasinya untuk industri kecil atau menengah serta digunakan untuk pengendalian teknologi tepat guna. Namun demikian umumnya PLC dibuat oleh perusahaan raksasa elektronika dari negara berteknologi tinggi seperti Amerika, Jerman, Jepang dan sebagainya yang relatif cukup sulit untuk dikuasainya.

Oleh karena itu tenaga kerja yang menangani PLC harus dituntut untuk menguasai teknologi PLC sehingga jika terjadi gangguan dapat segera mengatasinya, tanpa harus memanggil ahli dari perusahaan yang memproduksi PLC tersebut.

Melihat berkembangnya aplikasi PLC di dunia industri dan dunia usaha seperti tersebut di atas, maka Jurusan PT Elektro FT UNY, yang merupakan lembaga pendidikan yang akan menghasilkan lulusan untuk bekerja menjadi guru teknik elektro atau bekerja di dunia industri dan dunia usaha, dituntut untuk membekali kompetensi PLC bagi para mahasiswanya. Pemberian kompetensi PLC khususnya bagi mahasiswa adalah sangat penting, agar ketika lulus nantinya tidak gagap teknologi PLC, dan dapat menangani pekerjaan-pekerjaan yang berhubungan dengan kemajuan teknologi PLC di dunia pendidikan, di dunia industri dan dunia usaha.

B. Pengertian *Programmable Logic Controllers (PLC)*.

Programmable logic controllers (PLC) atau juga disebut *Programmable Controllers*, merupakan peralatan yang termasuk dalam keluarga komputer, yang menggunakan rangkaian terintegrasi untuk melaksanakan fungsi control. PLC mampu memberikan instruksi seperti sekuensial, pewaktuan, perhitungan, manipulasi data, dan komunikasi data untuk mengendalikan mesin atau proses industry. Hal ini berarti pula bahwa PLC sebagai komputer yang digunakan di industri yang arsitekturnya dirancang secara khusus dalam rangkaian central unit dan antar muka menuju peralatan dunia luar melalui unit input dan unit out put untuk mengendalikan peralatan industry secara praktis (L.A. Bryan and E.A. Bryan, 1997).

Menurut Festo (2004: 1) *Programmable Logic Controller (PLC)* merupakan piranti elektronika digital yang menggunakan memori yang bisa diprogram sebagai penyimpanan internal dari sekumpulan instruksi dengan mengimplementasikan fungsi-fungsi tertentu, seperti logika, sekuensial, pewaktuan, perhitungan, dan aritmetika. PLC dimanfaatkan untuk mengendalikan berbagai jenis mesin ataupun proses melalui modul I/O digital dan atau analog. Sesuai dengan namanya, konsep *Programmable Logic Controller (PLC)* diantaranya adalah :

1. *Programmable*, menunjukkan kemampuannya yang dapat dengan mudah diubah-ubah sesuai program yang dibuat dan kemampuannya dalam hal memory program yang telah dibuat.
2. *Logic*, menunjukkan kemampuannya dalam memproses input secara aritmetik (ALU), yakni melakukan operasi membandingkan, menjumlahkan, mengalikan, membagi, mengurangi, dan negasi.
3. *Controller*, menunjukkan kemampuan menghasilkan output yang diinginkan dalam mengontrol dan mengatur proses.

Secara umum PLC mempunyai fungsi antara lain sebagai berikut:

1. Kontrol Sekuensial, PLC memproses input sinyal biner menjadi output yang digunakan untuk keperluan pemrosesan teknik secara berurutan (sekuensial), di sini PLC menjaga agar semua step/langkah dalam proses sekuensial berlangsung dalam urutan yang tepat.
2. *Monitoring Plant*, PLC secara terus-menerus memonitor status suatu sistem (misalnya temperatur, tekanan, tingkat ketinggian) dan mengambil tindakan yang diperlukan sehubungan dengan proses yang dikontrol (misalnya nilai sudah melebihi batas) atau menampilkan pesan tersebut pada operator.

Berdasarkan uraian di atas menunjukkan bahwa PLC merupakan peralatan elektronika digital yang termasuk dalam keluarga komputer, yang menggunakan memori yang dapat diprogram sebagai penyimpanan internal dari sekumpulan instruksi dengan mengimplementasikan fungsi-fungsi tertentu, seperti logika, sekuensial, pewaktuan, perhitungan, dan aritmetika, untuk digunakan dalam pengendalian mesin atau proses di industri. PLC dapat memanipulasi, mengeksekusi, dan atau memonitor keadaan proses pada laju yang amat cepat, dengan dasar data yang bisa diprogram dalam sistem berbasis mikroprosesor integral. PLC menerima masukan dan menghasilkan keluaran sinyal-sinyal listrik untuk mengendalikan suatu sistem. Program dalam PLC pada dasarnya adalah pemrosesan sinyal-sinyal pada input dan output. Selain input dan output

pemrosesan dapat menggunakan internal memori atau flag. Hanya melalui input, controller dapat menerima informasi. Sedangkan melalui output, controller mengeluarkan data. Pada PLC, sinyal dari berbagai peralatan luar diinterfisi sehingga fleksibel dalam mewujudkan sistem kendali. Disamping itu, kemampuannya dalam komunikasi jaringan memungkinkan penerapan yang luas dalam berbagai operasi pengendalian sistem. Dalam sistem otomasi, PLC merupakan jantung dari system kendali. Dengan program yang disimpan dalam memori PLC, PLC dapat memonitor keadaan sistem melalui sinyal dari peralatan input, kemudian didasarkan atas logika program menentukan rangkaian aksi pengendalian peralatan output luar. PLC dapat digunakan untuk mengendalikan tugas-tugas sederhana yang berulang-ulang, atau di-interkoneksi dengan pengendali yang lain menggunakan komputer melalui sejenis jaringan komunikasi untuk mengintegrasikan pengendalian proses yang kompleks.

Menurut Hugh Jack (2005), PLC banyak digunakan untuk pengendalian di industry pada saat ini dan mendatang karena memiliki keunggulan, diantaranya sebagai berikut: (a) Biaya yang efektif untuk mengendalikan sistem yang kompleks; (b) Fleksibel dan dapat diterapkan kembali untuk mengendalikan sistem lain dengan cepat dan mudah; (c) Memiliki kemampuan komputasi yang dapat mengikuti pengendalian yang lebih canggih; (d) Lebih mudah dan memerlukan waktu yang singkat dalam melakukan penelusuran kerusakan atau gangguan; dan (e) Memiliki komponen yang handal yang memungkinkan dapat beroperasi selama bertahun-tahun. Dengan melihat keunggulan PLC seperti tersebut di atas, maka jika dibandingkan antara pengendalian menggunakan PLC dengan pengendalian menggunakan relay, maka pengendalian menggunakan PLC menawarkan berbagai kelebihan. Pada pengendalian menggunakan relay maka harus diberi pengawatan untuk melakukan fungsi khusus. Ketika sistem memerlukan perubahan, pengawatan relay harus diubah, dan dimodifikasi, yang memerlukan waktu. Adapun pengendalian yang menggunakan PLC dapat diprogram lebih sederhana dan sedikit pengawatan tangan. Pengontrol yang dapat diprogram juga menawarkan realibilitas *solid-state*, pemakaian daya yang lebih sedikit dan kemudahan untuk perluasan.

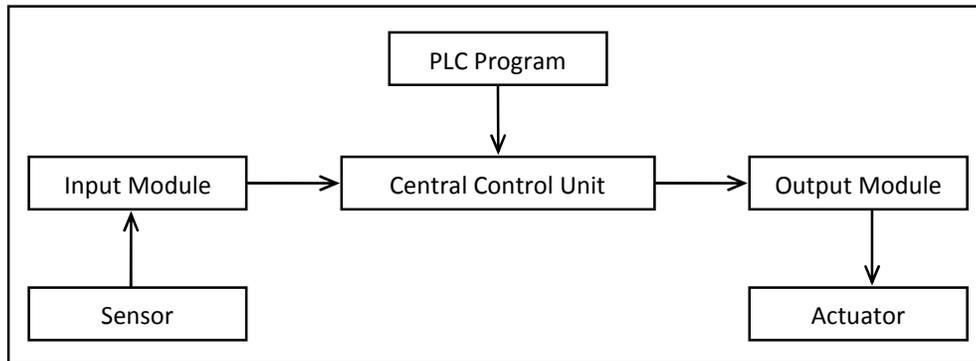
C. Bagian-bagian PLC.

Menurut Siemens (2005), Elemen dasar sebuah PLC antara lain modul masukan atau input, central processing unit (CPU), modul keluaran atau output, dan perangkat pemrograman. Jenis modul input yang digunakan oleh sebuah PLC tergantung pada jenis perangkat input yang digunakan. Modul input digital juga disebut input diskrit, termasuk di dalamnya adalah tombol on atau off. Modul input lainnya yaitu sinyal analog yang mewakili rentang nilai tegangan atau arus listrik. Fungsi utama dari input PLC adalah untuk mengkonversi sinyal yang disediakan oleh berbagai switch dan sensor menjadi sinyal logika yang dapat digunakan oleh CPU. CPU mengevaluasi status input, output, dan variabel lainnya dan mengeksekusi program. Selanjutnya CPU mengirimkan sinyal untuk memperbarui status output. Modul output mengubah sinyal kontrol dari CPU ke nilai digital atau analog yang dapat digunakan untuk mengontrol berbagai perangkat output. Perangkat pemrograman dipergunakan untuk memasukkan atau mengubah program PLC atau untuk memantau atau mengubah nilai-nilai yang telah tersimpan sebelumnya. Program dan variabel yang terkait disimpan dalam CPU.

Sistem input atau output diskret pada dasarnya merupakan antarmuka yang mengkoneksikan *central processing unit* (CPU) dengan peralatan input atau output luar. Lewat sensor-sensor yang terhubung dengan modul ini, PLC mengindra besaran-besaran fisik (posisi, gerakan, level, arus, tegangan) yang terasosiasi dengan sebuah proses atau mesin. Berdasarkan status dari input dan program yang tersimpan di memori PLC, CPU mengontrol perangkat luar yang terhubung dengan modul output. Secara fisik rangkaian input atau output dengan unit CPU tersebut terpisah secara kelistrikan, hal ini untuk menjaga agar kerusakan pada peralatan input atau output tidak menyebabkan hubungan singkat pada unit CPU. Isolasi rangkaian modul dari CPU ini umumnya menggunakan rangkaian optocoupler (Iwan Setiawan, 2006).

Secara lebih sederhana, menurut Festo (2004), PLC mempunyai komponen utama berupa CCU (*Central Control Unit*), Unit I/O, *Programming Console*, *Rack* atau

Mounting Assembly dan catu daya yang blok diagramnya seperti ditunjukkan pada Gambar 1.



Gambar 1. Komponen sistem PLC

1. *Central Control Unit (CCU)*

Central control unit merupakan unit pusat pengolah data yang digunakan untuk melakukan proses pengolahan data dalam PLC. Sentral kontrol unit merupakan mikroprosesor. Mikroprosesor terdiri dari Aritmatik dan Logic Unit (ALU), kontrol unit dan sejumlah kecil memori unit yang disebut register. Tugas dari ALU adalah untuk melakukan operasi aritmatika (penjumlahan, perkalian, dan sebagainya) dan logika (OR, AND, NOT, dan sebagainya).

2. Modul Input dan Modul Output.

Fungsi dari sebuah modul input adalah untuk mengubah sinyal masukan dari sensor ke PLC untuk diproses dibagian CCU. Sedangkan modul output digunakan untuk mengubah sinyal PLC ke dalam sinyal yang sesuai untuk menggerakkan aktuator. Dari modul input dan output kita dapat menentukan jenis suatu PLC dan modular PLC. Compact PLC terdiri dari input modul CCU dan output modul yang dikemas menjadi satu.

a. Modul input

Merupakan modul yang menghubungkan sensor-sensor dengan modul itu sendiri. Sinyal dari sensor tersebut akan dilanjutkan ke CCU. Fungsi penting dari modul input, diantaranya adalah:

- 1) Mendeteksi sinyal masukan.
- 2) Mengatur tegangan kontrol untuk batas tegangan logika masukan yang diijinkan.
- 3) Melindungi peralatan elektronik yang sensitif terhadap tegangan luar.
- 4) Menampilkan sinyal masukan.

b. Modul output

Modul output mengeluarkan sinyal dari CCU ke pengontrol yang diperlukan untuk menggerakkan aktuator sesuai dengan perintah yang diberikan. Fungsi dari modul output adalah sebagai berikut :

- 1) Mengatur tegangan kontrol untuk batas tegangan logika yang diijinkan.
- 2) Melindungi peralatan elektronik yang sensitif dari pengaruh luar.
- 3) Memberikan perlindungan adanya hubung singkat dan beban lebih.
- 4) Memberikan penguatan sinyal output sebelum dikeluarkan sehingga cukup kuat untuk menggerakkan aktuator.

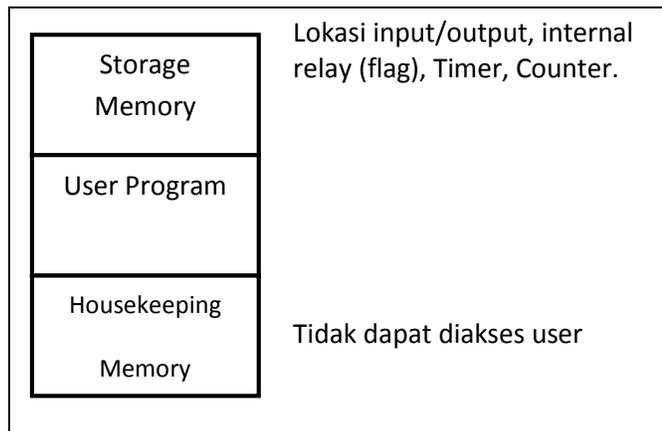
3. Catu Daya

Sistem PLC memerlukan dua buah catu daya. Satu untuk keperluan output sedangkan yang satunya untuk keperluan modul-modul itu sendiri yang menggunakan arus DC. Arus DC ini dapat diperoleh dari rangkaian terintegrasi atau transistor. Apabila sistem catu daya menggunakan IC TTL maka dapat menghasilkan tegangan 5 volt, tetapi bila menggunakan IC CMOS tegangan yang didapat akan bervariasi antara 3 sampai 18 volt.

D. Pemrograman PLC.

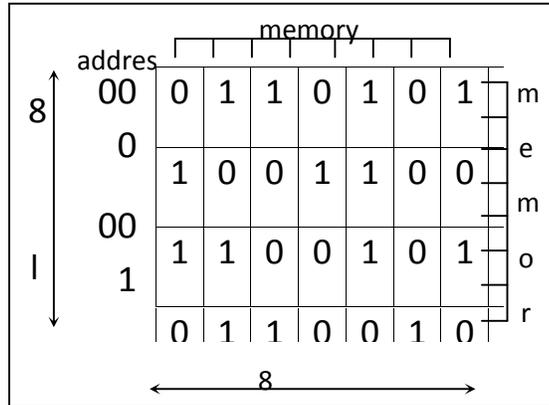
Menurut Festo (2004: 20-24), pada pemrograman PLC, program harus didesain secara sistematis, terstruktur dengan baik dan harus terdokumentasi supaya bebas dari

kesalahan, pemeliharaan mudah, efektif dalam pembiayaan. Untuk dapat memprogram PLC diperlukan sedikit pengetahuan tentang organisasi memori dari sebuah PLC. Memori dari PLC dibagi dalam tiga area, yaitu *storage memory*, *user memory*, dan *house keeping memory*. *Storage memory* digunakan untuk menyimpan input/output peralatan, *timer/counter* dan *preset value*-nya, *bit* atau *word* untuk internal relay pada lokasi memori tertentu. Area memori *user program* digunakan oleh user untuk melakukan programnya agar PLC dapat bekerja sebagaimana mestinya. Program user disimpan dalam jumlah 16 bit word, yang masing-masing word berisi satu instruksi program. Bit-bit dalam instruksi word dapat berupa “on” atau “off”. *Housekeeping memory* adalah area memori yang tidak dapat digunakan oleh *user*. Memori ini digunakan selama PLC beroperasi sebagai tempat mengerjakan fungsi-fungsi tertentu yang diperlukan membuat PLC dapat bekerja, untuk fungsi aritmatika dan mengerjakan fungsi operasi internal yang lain. Adapun pembagian memori PLC seperti ditunjukkan pada Gambar 2.



Gambar 2. Pembagian memori PLC

Data biner yang tidak segera dikehendaki untuk diproses harus diletakkan dalam penyimpanan sementara untuk dipakai kemudian. Instruksi program dan hasilnya disimpan dalam program memori (RAM). Sedangkan program dan data yang tidak berubah disimpan dalam *Read Only Memory* (ROM). Adapun susnan data memori PLC ditunjukkan seperti pada Gambar 3 berikut ini.



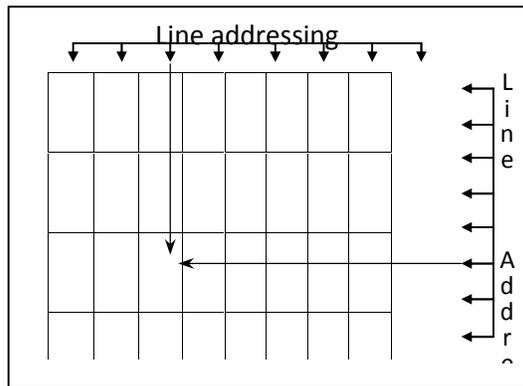
Gambar 3. Susunan data memori

Elemen terkecil dalam memori adalah lokasi memori. Hal ini dapat menerima unit daya, yaitu Bit (1 atau 0). Beberapa lokasi memori berasal dari sel memori, beberapa sel memori adalah blok memori. Setiap lokasi memori bisa terdapat 1 atau 0. Sebuah sel memori dapat menyimpan satu karakter yang terdiri dari 8 bit atau 1 byte.

Dalam pengalamatan memori, terdapat dua metode untuk mengalami dalam proses baca dan tulis, yaitu :

1. Pengalamatan secara bit. Proses baca dan tulis diakses secara *individu* lokasi memori. Pengalamatannya memerlukan keterangan mengenai alamat *line* dan kolom.
2. Pengalamatan secara word. Proses baca dan tulis : isi dari sel memori (*word data* = beberapa bit) ditulis atau dibaca secara bersama-sama. Program memori selalu dalam organisasi *word*, sebagai instruksi terdiri dari jumlah bit yang dibaca secara serentak.

Pengalamatan dibutuhkan untuk menemukan lokasi memori yang spesifik atau sel memori. Pada uraian di atas telah disebutkan bahwa ada 2 metode pengalamatan yaitu pengalamatan secara bit dan pengalamatan secara *word*. Dalam kasus pertama baris dan kolom harus diberikan sedangkan kasus kedua cukup pada baris alamat. Unit informasi terkecil adalah bit (1 atau 0). 1 kilobyte = 2^{10} byte = 1024 byte. Pengalamatan memori PLC ditunjukkan seperti Gambar 4 berikut ini.



Gambar 4. Pengalamatan memori PLC

Pada pembuatan program PLC, program ditulis dengan berbagai prosedur yang digunakan untuk menyelesaikan suatu permasalahan mengenai kontrol.

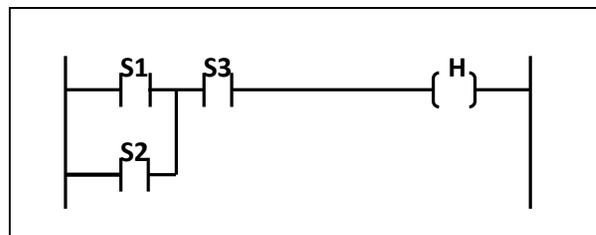
1. Langkah 1: Mempertimbangkan gambaran masalah. Definisi permasalahan harus menjabarkan problema kontrol secara tepat dalam bentuk yang detail. Informasi yang diperlukan seperti skema posisi, skema *sequence* dan tabel kebenaran menerangkan hubungan antara masukan dan keluaran dan juga berguna untuk tes terhadap resiko pada saat instalasi.
2. Langkah 2: *Allocation list*. Dalam hal ini berisi kondisi-kondisi program, termasuk *identifier* yang dipakai oleh keluaran atau masukan, juga nama yang singkat untuk *identifier* tersebut yang bersesuaian dengan alamat dari sinyal masukan atau keluaran.
3. Langkah 3: Pembuatan program. Masalah kontrol yang digambarkan disini adalah dalam bentuk abstrak.
4. Langkah 4: Pindahkan kedalam *controller*.
 Dalam tahap ini program yang telah siap dan lengkap diterjemahkan ke dalam bahasa mesin agar dapat dimengerti atau dibaca oleh CCU.

Pada bahasa pemrograman, terdapat banyak pilihan bahasa untuk membuat program dalam PLC. Masing-masing bahasa mempunyai keuntungan dan kerugian

tergantung dari sudut pandang kita sebagai *user* atau pemrogram. *Ladder Diagram* (LDR) adalah bahasa yang dimiliki oleh setiap PLC. Menurut Greg P. Zimmerman (2008), ladder logic atau Ladder diagram merupakan bahasa pemrograman yang paling umum digunakan dalam memprogram PLC. Ladder diagram merupakan suatu pendekatan pemrograman pertama kali yang digunakan untuk membuat program PLC, yang mengikuti bentuk diagram relay yang umum digunakan dalam bidang ketenagalistrikan. Simbol-simbol yang digunakan dalam ladder diagram relay umumnya terdiri dari garis rel kiri, garis rel kanan dan komponen rangkaian yang menghubungkan garis rel kiri dan garis rel kanan. Logika dari setiap sirkuit diselesaikan dari kiri ke kanan. Selain itu ada beberapa jenis PLC yang mendukung bahasa pemrograman lain seperti *ladder diagram*, *statement list*, *instruction list*, dan *function chart*.

1. *Ladder Diagram*

Ladder Diagram menggambarkan program berupa bentuk grafik. Diagram ini dikembangkan dari kontak-kontak relay yang tersruktur yang menggambarkan aliran arus listrik. Dalam ladder diagram, terdapat dua garis vertikal dimana garis vertikal sebelah kiri dihubungkan dengan sumber tegangan positif catu daya sedangkan garis vertikal sebelah kanan dihubungkan dengan sumber tegangan negatif catu daya. Di antara dua garis ini dipasang kontak-kontak yang menggambarkan kontrol dari *switch*, sensor, atau output. Salah satu baris dari diagram disebut dengan satu rung. Input menggunakan simbol “[]” (kontak, normal terbuka) dan “[/]” (negasi kontak, normal tertutup). Output mempunyai simbol “()” yang terletak paling kanan. Selama pemrograman setiap simbol yang diberikan adalah alamat PLC sesungguhnya atau merupakan alamat simbolik (misalnya : S1, S2, S3, H). Ladder diagram PLC seperti ditunjukkan Gambar 5 berikut ini.



Gambar 5. Ladder diagram

2. *Statemen list*

Statement list merupakan bahasa pemrograman tingkat tinggi. Semua hubungan logika dan kontrol sekuens dapat diprogram dengan menggunakan perintah dalam bahasa ini. Untuk membuat *statement list* dibutuhkan dukungan *allocation list*. *Allocation list* merupakan proses pengalamatan pada program yang meliputi *Absolute Operation* dan *Symbol Operation* dan *Commentary*. *Absolute Operation* merupakan perintah yang akan dijalankan oleh program. Sementara *Symbol Operation* berarti bahasa simbol untuk memudahkan pengenalan pada hardware yang akan bekerja. Sedangkan *Commentary* berfungsi untuk memperjelas kerja. *Alocation list* PLC seperti ditunjukkan pada Tabel 3 berikut ini.

Tabel 1. *Alocation list* PLC

Absolut Op.	Symbolic Op.	Deskripsi
I0.0	S1	Saklar "ON" / "1" saat ditekan
I0.1	S2	
I0.2	S3	
O0.0	H	"ON" saat bernilai "1"

3. *Instruction list*

Instruction list merupakan bentuk lain dari bahasa pemrograman PLC. Semua hubungan logika dan kontrol sekuens dapat diprogram dengan menggunakan perintah dalam *instruction list*. Perintah-perintahnya adalah dalam bentuk singkatan dan mempunyai arti yang khusus, sebagai contoh :

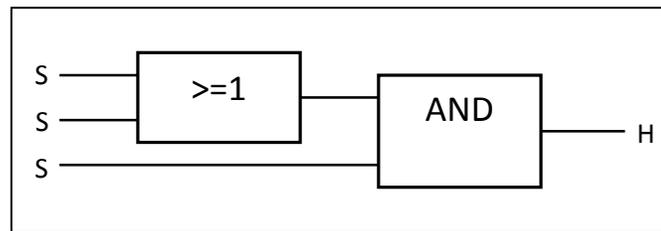
H ("Load")

A, O, N (“AND, OR, NOT”)

= (“Allocation”)

4. *Function Chart* (FCH)

Function chart diturunkan dari diagram logika dan merupakan gambaran program dalam bentuk grafik. Dapat digunakan untuk pemrograman logika yang sederhana. *Function chart* PLC seperti ditunjukkan Gambar 6 berikut ini.

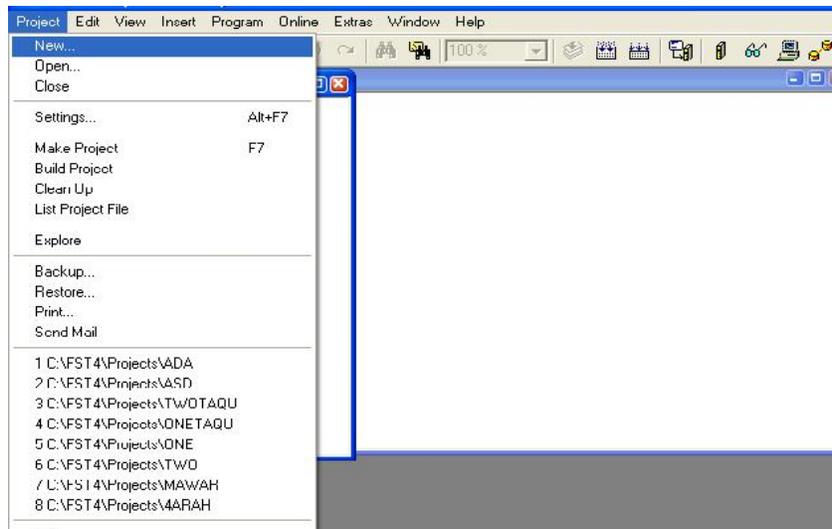


Gambar 6. *Function chart* PLC.

E. Contoh Pemrograman PLC Festo.

Menurut Sussanto Setiawan (2006), langkah-langkah dalam pembuatan program pada software Festo adalah seperti berikut ini.

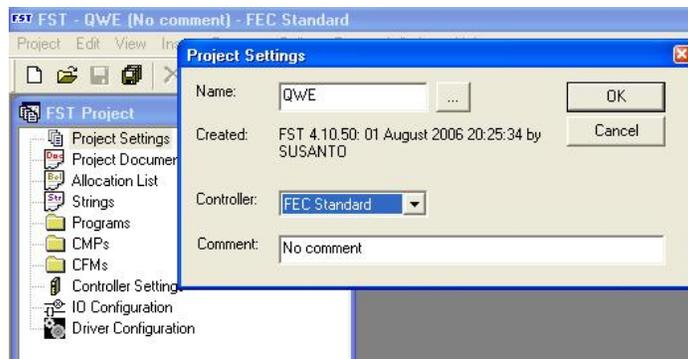
1. Menghidupkan komputer dan membuka software FESTO (FSTIPC)
2. Menekan (Klik) *Project* kemudian *New*



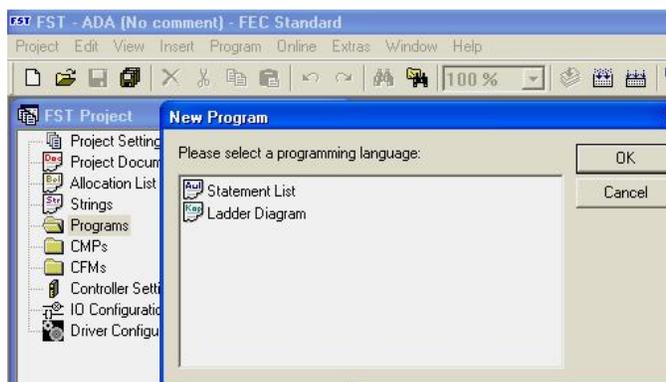
3. Memberi nama maksimal 8 karakter tanpa spasi dan mudah diingat



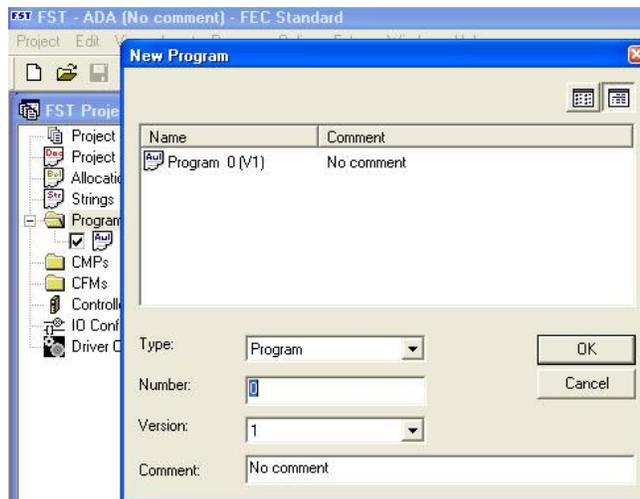
4. Setelah memberi nama pada *project*, kemudian akan muncul *Project setting* pada *controller* isilah dengan FEC standard dan pada *Comment* isilah terserah anda.



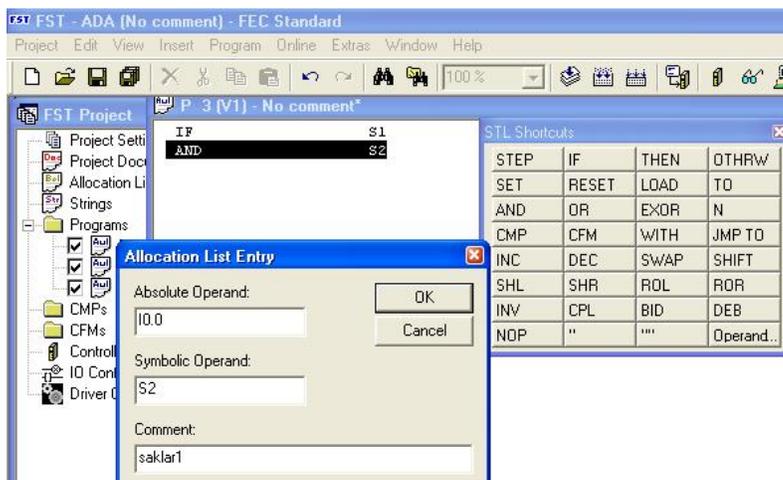
5. Klik kanan folder program pilih new program (Ctrl+ N) kemudian pilih bahasa pemrograman antara ladder diagram dan statement list yang paling dimengerti dan dipahami



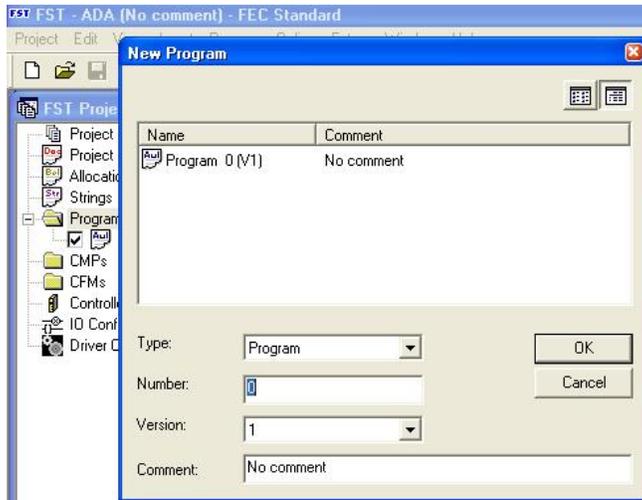
6. Ketika kita memilih Statement list maka tampilannya di bawah ini. Pada number isilah program keberapa dan pada comment dan version terserah kita yang terpenting harus berurutan.



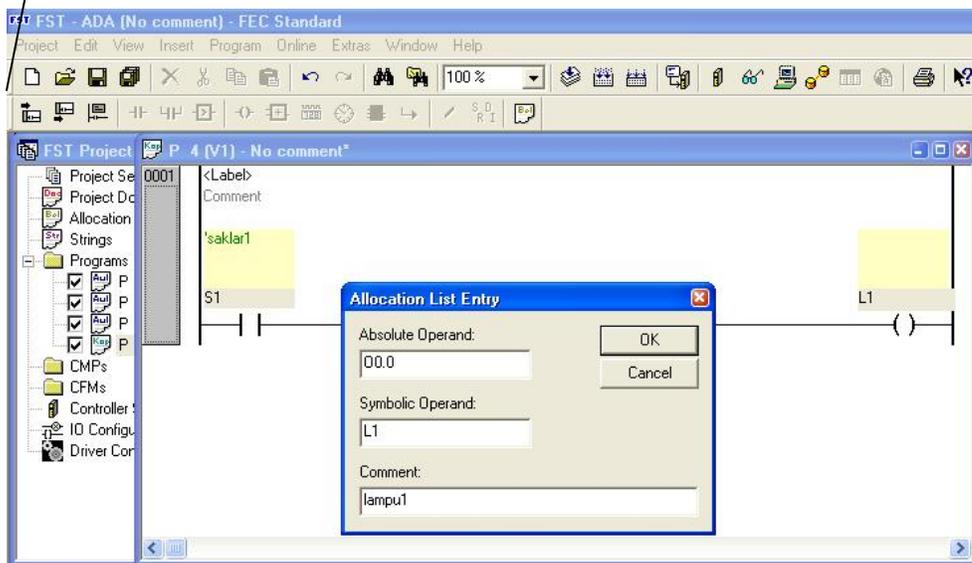
7. Tampilan dan cara membuat program dengan statement list.



8. Tampilan dan cara membuat program bila menggunakan ladder diagram. Pada number isilah program keberapa dan pada comment dan version terserah kita yang terpenting harus berurutan. Caranya kita tinggal klik pada shortcut seperti pada gambar dibawah ini.

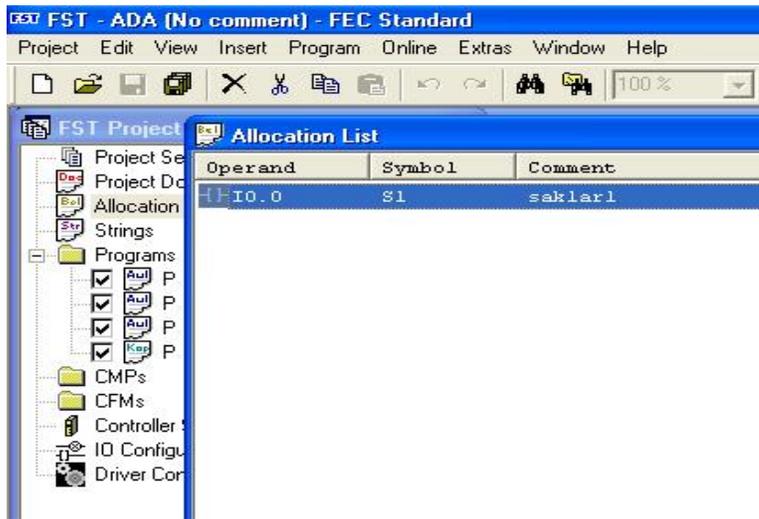


Deretan perintah operasi (Shortcut) dalam ladder diagram

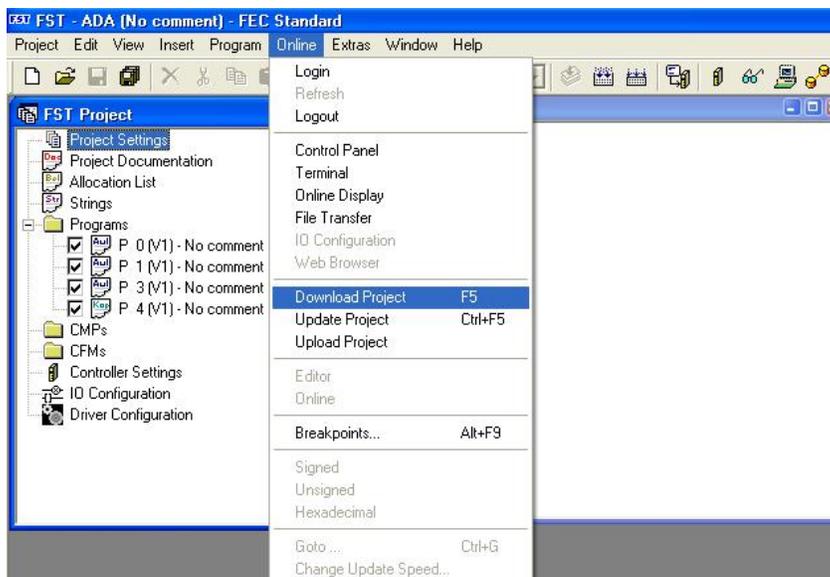


9. Pembuatan Allocation List biasanya langsung terjadi saat pembuatan program dengan statement list ataupun ladder diagram. Berikut caranya: klik kanan folder allocation List

kemudian open maka akan terlihat sudah ada datanya bila kita sudah membuat program dengan ladder diagram maupun statement list



10. Bila program yang kita buat sudah selesai maka kita akan mendownloadnya ke PLC dengan menekan tombol F5 pada keyboard berikut gambarannya.



11. Bila pendownloadan program sukses maka kita tinggal menjalankan PLC. Sedangkan bila masih salah cek program yang dibuat. Indikasi bahwa program salah bisa dilihat jika kita telah selesai mendownload pada kotak bagian bawah program dibawah start.

Lebih lanjut Sussanto Setiawan (2006) memberikan contoh pemrograman PLC Festo untuk pengendalian lampu lalu lintas seperti berikut ini.

PROGRAM ONE (KENDALI PLC 1)

Allocation List

O0.0 h1
O0.1 k1
O0.2 m1
O0.3 h2
O0.4 k2
O0.5 m2
O0.6 rampung1 tanda plc 1 sudah selesai
I0.0 start tombol start
I0.1 stop saklar stop
I0.6 rampung2 tanda plc 2 sudah selesai
F0.0 flag0
F0.1 flag1
P1
P2
P3
T0 timer0
T1 timer1
T2 timer2
T3 timer3
T4 timer4
T5 timer5

Program 0 (P 0)

```
IF           start     'tombol start
THEN RESET    P3
      SET     flag0
```

```

    RESET    P2
    SET      P1
IF      rampung2   'tanda plc 2 sudah selesai
    AND      flag0
THEN SET    P1
IF      stop      'saklar stop
THEN RESET P1
    RESET    P2
    SET      P3

```

Program 1 (P 1)

STEP 1

```

IF      flag0
    AND      rampung2   'tanda plc 2 sudah selesai
THEN SET    h1
    RESET    k1
    RESET    m1
    RESET    h2
    RESET    k2
    SET      m2
    SET      timer0
    WITH     15s

```

STEP 2

```

IF      N timer0
THEN RESET h1
    SET      k1
    RESET    m1
    RESET    h2
    RESET    k2
    SET      m2
    SET      timer1

```

```

        WITH      2s
STEP 3
IF      N  timer1
THEN RESET      h1
    RESET      k1
    SET        m1
    SET        h2
    RESET      k2
    RESET      m2
    SET        timer2
        WITH      15s
STEP 4
IF      N  timer2
THEN RESET      h1
    RESET      k1
    SET        m1
    RESET      h2
    SET        k2
    RESET      m2
    SET        timer3
        WITH      2s
STEP 5
IF      N  timer3
THEN RESET      h1
    RESET      k1
    SET        m1
    RESET      h2
    RESET      k2
    SET        m2
    SET        timer4

```

```

        WITH      2s
STEP 6
IF      N      timer4
THEN SET      rampung1      'tanda plc 1 sudah selesai
        SET      timer5
        WITH      0.5s
STEP 7
IF      N      timer5
THEN RESET      rampung1      'tanda plc 1 sudah selesai

```

Program 2 (P 2)

```

IF      flag0
THEN SET      m1
        RESET      k1
        RESET      h1
        SET      m2
        RESET      k2
        RESET      h2

```

Program 3 (P 3)

```

IF      flag0
THEN SET      m1
        RESET      k1
        RESET      h1
        SET      m2
        RESET      k2
        RESET      h2
        RESET      rampung1      'tanda plc 1 sudah selesai

```

PROGRAM TWO (kendali plc 2)

Program 0 (P 0)

```
IF          start      'tombol start
THEN SET    flag0
  RESET    P4
  SET      P3
  SET      P2
IF          rampung2   'tanda plc 1 sudah selesai
  AND      flag0
THEN RESET  P2
  RESET    P3
  SET      P1
IF          stop       'saklar stop
THEN RESET  P1
  RESET    P3
  RESET    P2
  SET      P4
```

Program 1 (P 1)

STEP 1

```
IF          flag0
  AND      rampung2   'tanda plc 1 sudah selesai
THEN SET    h1
  RESET    k1
  RESET    m1
  RESET    h2
  RESET    k2
  SET      m2
  SET      timer0
```

```

        WITH      15s
STEP 2
IF      N  timer0
THEN RESET      h1
    SET      k1
    RESET     m1
    RESET     h2
    RESET     k2
    SET      m2
    SET      timer1
        WITH      2s
STEP 3
IF      N  timer1
THEN RESET      h1
    RESET     k1
    SET      m1
    SET      h2
    RESET     k2
    RESET     m2
    SET      timer2
        WITH      15s
STEP 4
IF      N  timer2
THEN RESET      h1
    RESET     k1
    SET      m1
    RESET     h2
    SET      k2
    RESET     m2
    SET      timer3

```

```

        WITH      2s
STEP 5
IF      N      timer3
THEN RESET      h1
        RESET      k1
        SET      m1
        RESET      h2
        RESET      k2
        SET      m2
        SET      timer4
        WITH      2s
STEP 6
IF      N      timer4
THEN SET      rampung1      'tanda plc 2 sudah selesai
        SET      timer5
        WITH      0.5s
STEP 7
IF      N      timer5
THEN RESET      rampung1      'tanda plc 2 sudah selesai
Program 2 (P 2)
STEP 1
IF      flag0
        AND      start      'tombol start
THEN SET      rampung1      'tanda plc 2 sudah selesai
        SET      timer6
        WITH      0.2s
STEP 2
IF      N      timer6
THEN RESET      rampung1      'tanda plc 2 sudah selesai
Program 3 (P 3)

```

```

IF          flag0
THEN SET   m1
  RESET   k1
  RESET   h1
  SET     m2
  RESET   h2
  RESET   k2
  RESET   rampung1   'tanda plc 2 sudah selesai

```

Program 4 (P 4)

```

IF          flag0
THEN SET   m1
  RESET   k1
  RESET   h1
  SET     m2
  RESET   h2
  RESET   k2
  RESET   rampung1   'tanda plc 2 sudah selesai

```

Allocation List

```

O0.0    h1
O0.1    k1
O0.2    m1
O0.3    h2
O0.4    k2
O0.5    m2
O0.6    rampung1   tanda plc 2 sudah selesai
I0.0    start   tombol start
I0.1    stop   saklar stop
I0.6    rampung2   tanda plc 1 sudah selesai
F0.0    flag0
F0.1    flag1

```

P1

P2

P3

P4

T0 timer0

T1 timer1

T2 timer2

T3 timer3

T4 timer4

T5 timer5

T6 timer6

Daftar Pustaka

- Bryand L.A. and Bryan E.A. (1997). *Programmable Controllers Theory and Implementation Second Edition*. Georgia: Industrial Text Company.
- Festo. (2004). *Programmable Logic Controller*. Jakarta : PT Festo.
- Greg P. Zimmerman. (2008). *Programmable Logic Controllers and Ladder Logic*. South Dakota: Deaprtmen of Humanities South Dakota School of Mines and Technology.
- Hugh Jack. (2007). *Automating manufacturing System With PLCs*. Retrieved August 1, 2012 from the World Wide Web: <http://claymore.engineer.gvsu.edu/jackh/books.html>
- Iwan Setiawan. (2006). *Programmable Logic Controllers dan Teknik Perancangan Sistem Kontrol*. Yogyakarta: Penerbit Andi.
- Siemens. (2005). *The SIMATIC S7-200 Programmable Controllers*, Retrieved August 2, 2012 from the World Wide Web: <http://www.automation.siemens.com/en/s7-200/index.htm>
- Sussanto Setiawan. (2006). *Pengendalian Lampu Lalu lintas Menggunakan PLC*. Yogyakarta: Jurusan PT Elektro FT UNY.