

## **BAB II**

### **KAJIAN TEORITIK DAN KERANGKA BERPIKIR**

#### **2.1. Kajian Teoritik**

##### 2.1.1. Pengertian Prototipe

Menurut Kamus Besar Bahasa Indonesia Prototipe memiliki arti model yang mula (model asli) yang menjadi contoh atau standar ukuran dari sebuah entitas. Dalam bidang desain, sebuah prototipe dibuat sebelum dikembangkan atau justru dibuat khusus untuk pengembangan sebelum dibuat dalam skala sebenarnya atau sebelum diproduksi secara massal.

Prototipe bisa diartikan juga sebagai bentuk awalnya saja dan tidak menutup kemungkinan bisa dikembangkan menjadi skala yang lebih besar<sup>1</sup>. Jadi jangan heran apabila banyak prototipe yang dibuat dengan ukuran yang sangat kecil, ini bertujuan untuk membuat sebuah model awal program rancangan perangkat-perangkat ataupun sebuah sistem. Terdapat 3 pendekatan utama prototipe, yaitu :

##### a. *Throw-Away*

Prototipe dibuat dan dites. Pengalaman yang diperoleh dari pembuatan prototipe digunakan untuk membuat produk akhir (*final*), kemudian prototipe tersebut dibuang (tidak dipakai).

---

<sup>1</sup>Ian Sommerville, *Software Engineering (Rekayasa Perangkat Lunak)*, Erlangga, Jakarta, hlm. 175.

b. *Incremental*

Produk akhirnya dibuat sebagai komponen-komponen yang terpisah. Desain produk akhirnya secara keseluruhan hanya ada satu tetapi dibagi dalam komponen-komponen lebih kecil yang terpisah (*independent*).

c. *Evolutionary*

Pada metode ini, prototipenya tidak dibuang tetapi digunakan untuk iterasi desain berikutnya. Dalam hal ini, sistem atau produk yang sebenarnya dipandang sebagai evolusi dari versi awal yang sangat terbatas menuju produk *final* atau produk akhir.

### 2.1.2. Pengertian Kincir/Bianglala

Kincir angin adalah sebuah alat yang mampu memanfaatkan kekuatan angin untuk dirubah menjadi kekuatan mekanik. Dari proses itu memberikan kemudahan berbagai kegiatan manusia yang memerlukan tenaga yang besar seperti memompa air untuk mengairi sawah atau menggiling biji-bijian. Kincir angin modern adalah mesin yang digunakan untuk menghasilkan energi listrik, disebut juga dengan turbin angin. Turbin angin kebanyakan ditemukan di Eropa dan Amerika Utara<sup>2</sup>.

---

<sup>2</sup>[http://id.wikipedia.org/wiki/Kincir\\_angin](http://id.wikipedia.org/wiki/Kincir_angin)



**Gambar 2.1. Kincir Angin**

Hampir seluruh penjuru negeri saat ini telah familier dengan mainan khas yang satu ini. “**Bianglala**”, sebut saja seperti itu lantaran namanya memang akrab disapa demikian oleh masyarakat. Biang lala boleh dibilang mengadopsi sistem kerja Kincir Angin dengan menggunakan sebuah motor pada poros tengah alat tersebut.



**Gambar 2.2 Bianglala**

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

#### 2.1.3.1. Sejarah PLC

PLC pertama kali diperkenalkan pada tahun 1960-an. Alasan utama perancangan PLC adalah untuk menghilangkan beban ongkos perawatan dan penggantian sistem kontrol mesin berbasis relai. Bedford Associate (Bedford, MA) mengajukan usulan yang diberi nama MODICON (kepanjangan *Modular Digital controller*) untuk perusahaan-perusahaan mobil di Amerika. Sedangkan perusahaan lain mengajukan sistem berbasis komputer (PDP-8). MODICON 084 merupakan PLC pertama didunia.

Saat kebutuhan produksi berubah maka demikian pula dengan sistem kontrolnya. Hal ini menjadi sangat mahal jika perubahannya terlalu sering. Karena relai merupakan alat mekanik, maka relai memiliki umur hidup atau masa penggunaan yang terbatas, yang akhirnya membutuhkan jadwal perawatan yang ketat. Pelacakan kerusakan atau kesalahan menjadi cukup membosankan jika banyak relai yang digunakan. Bayangkan saja sebuah panel kontrol yang dilengkapi dengan monitor ratusan hingga ribuan relai yang terkandung pada sistem kontrol tersebut. Bagaimana kompleksnya melakukan pengkabelan pada relai-relai tersebut. Dengan demikian, pengontrol baru ini harus memudahkan para teknisi perawatan dan teknisi lapangan melakukan pemrograman. Umur alat harus menjadi lebih panjang dan program proses dapat dimodifikasi atau dirubah

dengan lebih mudah, Serta harus mampu bertahan dalam lingkungan industri yang keras.

Pada pertengahan tahun 1970-an, teknologi PLC yang dominan adalah sekuenser mesin-kondisi dan CPU berbasis *bit-slice*. Prosesor AMD 2901 dan 2903 cukup populer digunakan dalam MODICON dan PLC A-B. Mikroprosesor konvensional kekurangan daya dalam menyelesaikan logika PLC secara cepat, kecuali PLC kecil. Setelah mikroprosesor konvensional mengalami perbaikan dan pengembangan, PLC konvensional mulai banyak menggunakannya. Walaupun begitu, hingga saat ini ada yang masih berbasis pada AMD 2903.

Kemampuan komunikasi pada PLC mulai muncul pada awal-awal tahun 1973. Sistem yang pertama adalah *Modbus*-nya MODICON. Dengan demikian PLC bisa berkomunikasi dengan PLC lain dan bisa ditempatkan lebih jauh dari lokasi mesin sesungguhnya yang dikontrol. Sayangnya, kurangnya standarisasi mengakibatkan komunikasi PLC menjadi mimpi buruk untuk protokol-protokol dan jaringa-jaringan yang tidak kompatibel. Tetapi bagaimanapun juga, saat itu merupakan tahun yang hebat untuk PLC.

Pada tahun 1980-an dilakukan usaha untuk menstandarisasi komunikasi dengan protokol otomasi pabrik milik General Motor MAP (*Manufacturing Automation Protocol*). Saat itu juga merupakan waktu untuk memperkecil ukuran PLC dan pembuatan perangkat lunak pemrograman melalui pemrograman simbolik dengan

komputer PC daripada terminal pemrogram atau penggunaan pemrogram genggam (*handheld programmer*). Sekarang PLC terkecil seukuran dengan sebuah kontrol relai tunggal. Pada tahun 1990-an dilakukan reduksi protokol baru dan modernisasi lapisan fisik dari protokol-protokol populer yang bertahan sampai dengan tahun 1980-an. Standar terakhir (IEC 1131-3).<sup>3</sup>

Seiring perkembangan teknologi *solid state*, saat ini PLC telah mengalami perkembangan luar biasa, baik dari ukuran, kepadatan komponen serta dari segi fungsionalnya. Beberapa peningkatan perangkat keras dan perangkat lunak ini diantaranya adalah:

- a. Ukuran semakin kecil dan kompak.
- b. Jumlah *input-output* yang semakin banyak dan padat
- c. Waktu eksekusi program yang semakin cepat.
- d. Pemrograman relatif semakin mudah. Hal ini terkait dengan perangkat lunak pemrograman yang semakin *user friendly*.
- e. Memiliki kemampuan komunikasi dan sistem dokumentasi yang semakin baik.
- f. Jenis instruksi/fungsi semakin banyak dan lengkap.

---

<sup>3</sup>Rista Aprilowena, 2012. Sejarah PLC, <https://vavarivistava.wordpress.com/2012/12/06/plc/>, terakhir diakses 8 Agustus 2015, jam 20.22 WIB.

### 2.1.3.2. Pengertian PLC

*Programmable Logic Controllers* (PLC) adalah komputer elektronik yang mudah digunakan dan memiliki fungsi kendali untuk berbagai tipe dan tingkat kesulitan yang beraneka ragam. Definisi *Programmable Logic Controller* menurut Capiel (1982) adalah “sistem elektronik yang beroperasi secara digital dan didesain untuk pemakaian di lingkungan industri, dimana sistem ini menggunakan memori yang dapat diprogram untuk penyimpanan secara internal instruksi-instruksi yang mengimplementasikan fungsi-fungsi spesifik seperti logika, urutan, perwaktuan, pencacahan dan operasi aritmatik untuk mengontrol mesin atau proses melalui modul-modul I/O digital maupun analog”.

Berdasarkan namanya, konsep PLC adalah sebagai berikut :

a. *Programmable*

Menunjukkan kemampuan dalam hal memori untuk menyimpan program yang telah dibuat yang dengan mudah diubah-ubah fungsi atau kegunaannya.

b. *Logic*

Menunjukkan kemampuan dalam memproses *input* secara aritmatik dan *logic* yakni melakukan operasi membandingkan, menjumlahkan, mengalikan, membagi, mengurangi, negasi, AND, OR, dan lain sebagainya.

c. *Controller*

Menunjukkan kemampuan dalam mengontrol dan mengatur proses sehingga menghasilkan *output* yang diinginkan.

PLC dirancang untuk menggantikan suatu rangkaian relai sekuensial dalam suatu sistem kontrol. Selain dapat diprogram, alat ini juga dapat dikendalikan, dan dioperasikan oleh orang yang tidak memiliki pengetahuan di bidang pengoperasian komputer secara khusus. PLC ini memiliki bahasa pemrograman yang mudah dipahami dan dapat dioperasikan bila program yang telah dibuat dengan menggunakan software yang sesuai dengan jenis PLC yang digunakan sudah dimasukkan. Jenis salah satu PLC yang digunakan dapat dilihat pada gambar 2.3.



**Gambar 2.3. PLC Omron Sysmac CP1E**



Alat ini bekerja berdasarkan input-input yang ada dan tergantung dari keadaan pada suatu waktu tertentu yang kemudian akan meng-*on* atau meng-*off* kan *output-output*. 1 menunjukkan bahwa keadaan yang diharapkan terpenuhi sedangkan 0 berarti keadaan yang diharapkan tidak terpenuhi. PLC juga dapat diterapkan untuk pengendalian sistem yang memiliki *output* banyak.

#### 2.1.3.3. Fungsi PLC

Fungsi dan kegunaan PLC sangat luas. Praktiknya, PLC dapat dibagi secara umum dan secara khusus. Secara umum fungsi PLC adalah sebagai berikut :

a. *Sekuensial Control*

PLC memproses *input* sinyal biner menjadi *output* yang digunakan untuk keperluan pemrosesan teknik secara berurutan. PLC menjaga agar semua langkah dalam proses berlangsung dalam urutan yang tepat.

b. *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.

Sedangkan fungsi PLC secara khusus adalah dapat memberikan *input* ke CNC (*Computerized Numerical Control*). Beberapa PLC dapat memberikan input ke CNC untuk kepentingan pemrosesan lebih lanjut. CNC bila dibandingkan dengan PLC mempunyai ketelitian yang lebih tinggi dan lebih mahal harganya. CNC biasanya dipakai untuk proses *finishing*, membentuk benda kerja, *moulding* dan sebagainya.<sup>4</sup>Keuntungan menggunakan PLC adalah sebagai berikut:

a. Fleksibel

PLC dapat mengontrol lebih dari satu alat dan programnya mudah dimodifikasi dalam jangka waktu yang relatif singkat.

b. Jumlah kontak bantu tak terbatas

PLC mempunyai jumlah kontak yang banyak untuk setiap koil dalam pemrogramannya, dapat mencapai ratusan kontak untuk satu kontaktor, tergantung dari kapasitas memori.

c. Mudah diamati

Operasi program PLC dapat dilihat selama operasi pada layar monitor. Jika operasi mengalami kesalahan maka dapat diketahui. Hal ini dikarenakan rangkaian logika terlihat terang pada layar monitor jika mendapat tegangan/terhubung.

---

<sup>4</sup>Rista Aprilowena, 2012. Sejarah PLC, <https://vavarivistava.wordpress.com/2012/12/06/plc/>, terakhir diakses 8 Agustus 2015, jam 20.25 WIB.

d. Kecepatan operasi

Kecepatan operasi program PLC sangat cepat. Kecepatan operasi logika PLC dibatasi oleh *scan time*, yang membutuhkan beberapa mili/detik.

e. Metode pemrograman dengan diagram tangga

Pemrograman *ladder diagram* PLC dapat diselesaikan dalam model diagram tangga bukan dalam bentuk *teks* sehingga lebih mudah untuk dilihat dan dipahami.

#### 2.1.3.4. Prinsip Kerja PLC

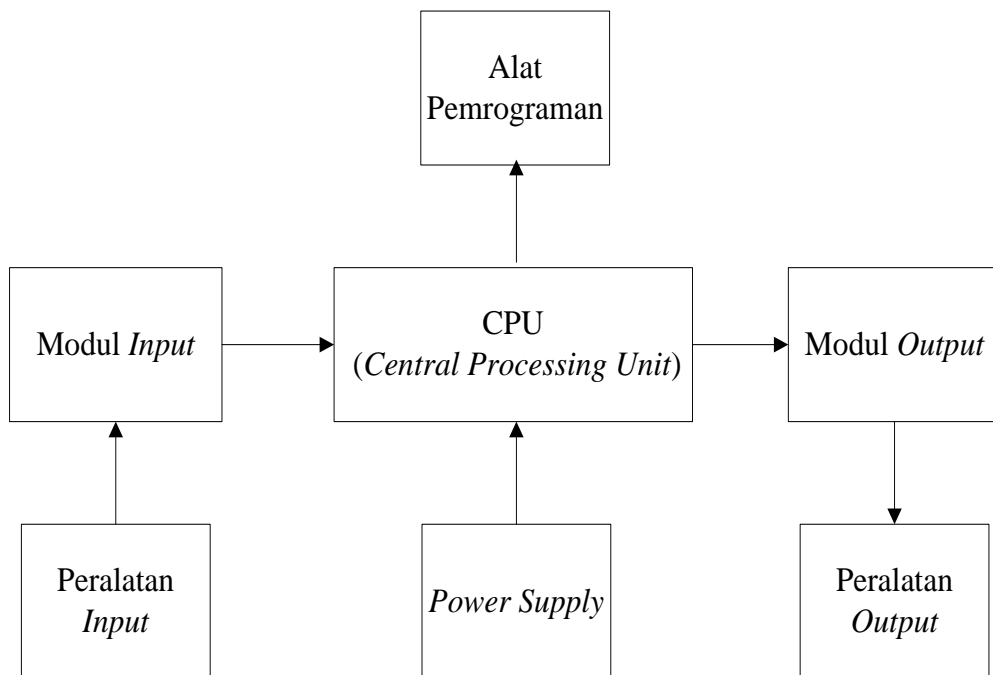
Prinsip kerja PLC dapat diuraikan sebagai peralatan elektronik yang dibangun dari *mikroprosesor* untuk memonitor keadaan dari peralatan *input* dan kemudian dianalisa sesuai dengan kebutuhan perencanaan (*programmer*) untuk mengontrol keadaan *output*.<sup>5</sup>

Pada prinsipnya PLC bekerja dengan cara menerima data-data dari peralatan *input* luar seperti pada gambar 2.4. Peralatan *input* dapat berupa saklar, tombol, dan sensor. Data-data yang masuk dari peralatan input ini berupa sinyal diskrit atau *analog*. Modul *input* ini akan mengidentifikasi serta mengubah sinyal yang masuk tersebut ke dalam bentuk tegangan yang sesuai oleh CPU sehingga menjadi sinyal-sinyal digital. Kemudian oleh CPU yang ada di dalam PLC, sinyal-sinyal digital tersebut akan diolah berdasarkan program yang

---

<sup>5</sup>Handy Wicaksono, *Programmable Logic Controller (Teori, Pemrograman dan Aplikasinya dalam Otomasi Sistem)*, Graha Ilmu, Yogyakarta, hlm 50.

telah disimpan dalam memori dan selanjutnya sinyal tersebut dikirim ke modul *output*. Bentuk sinyal digital ini akan diubah oleh modul *output* menjadi sinyal yang dapat digunakan untuk menjalankan peralatan *output* yang dapat berupa lampu, katup, motor, kontaktor, ataupun *relay*. Peralatan *output* inilah yang nantinya akan mengoperasikan sistem atau proses yang akan dikontrol.



**Gambar 2.4. Arsitektur PLC**

Sumber: dokumentasi

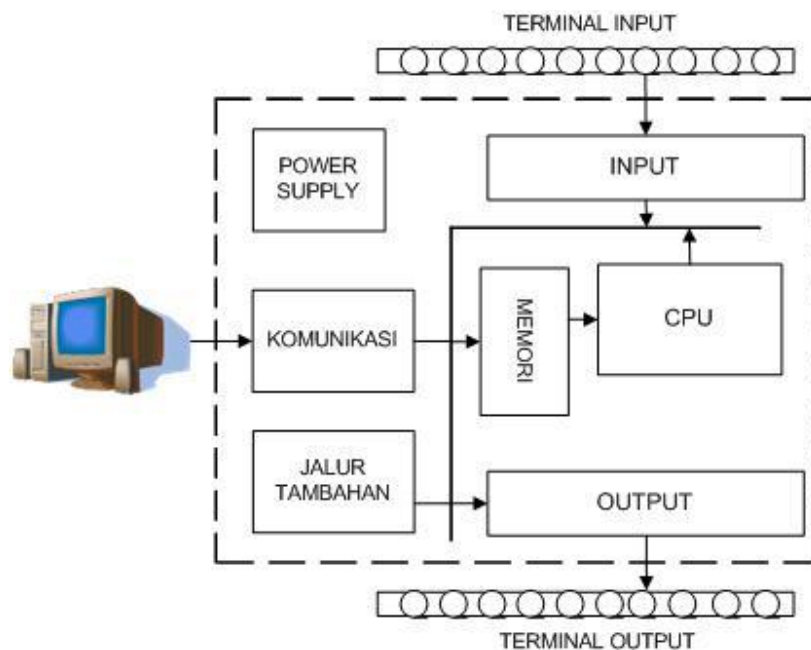
Sebuah PLC berisi CPU (*Central Processing Unit*) atau otak dari PLC yang berisi sebuah aplikasi program, modul *interface input* dan *output* yang terhubung secara langsung ke *field I/O devices* dan lebih lengkapnya PLC dapat diartikan sebuah alat kontrol yang bekerja berdasarkan pada pemrograman dan eksekusi instruksi logika. PLC beroperasi dengan cara memeriksa *input* dari sebuah proses guna mengetahui statusnya kemudian sinyal *input* ini diproses berdasarkan instruksi logika yang telah diprogram dalam memori.

Pada PLC juga dipersiapkan *internal input* dan *output* untuk proses dalam PLC sesuai dengan kebutuhan program, Dimana *internal input* dan *output* ini hanya sebagai *flag* dalam proses. Di dalam PLC juga dipersiapkan *timer* yang dapat dibuat dalam konfigurasi *on delay*, *off delay*, *on timer*, *off timer* dan lain- lain sesuai dengan programnya.

Untuk memproses, *timer* PLC tersebut memanggil berdasarkan alamatnya. Untuk melaksanakan fungsi sebagai kontrol sistem, PLC didukung oleh perangkat lunak yang merupakan bagian penting dari program PLC dan sebagai hasilnya berupa sinyal *output*. Sinyal *output* inilah yang dipakai untuk mengendalikan peralatan atau mesin. Antarmuka (*interface*) yang terpasang di PLC memungkinkan PLC dihubungkan secara langsung ke *actuator* atau *transducer* tanpa memerlukan *relay*.

### 2.1.3.5. Komponen Pada PLC

Pada kenyataannya PLC merupakan suatu sistem mikrokontroler yang digunakan untuk keperluan industri. PLC dapat dikatakan sebagai suatu perangkat keras dan lunak yang dapat dibuat untuk diaplikasikan dalam dunia industri.<sup>6</sup> Secara umum PLC memiliki bagian-bagian yang sama dengan komputer maupun mikrokontroler, yaitu CPU, Memori dan I/O. Susunan komponen PLC dapat dilihat pada Gambar 2.5.



**Gambar 2.5. Komponen Pada PLC**

Sumber: Husanto, Thomas, ST, MT, PLC (*Programmable Logic Controller*) FP Sigma, C.V Andi Offse, Yogyakarta.

<sup>6</sup>Husanto, Thomas, ST, MT, "PLC (*Programmable Logic Controller*) FP Sigma". C.V Andi, Yogyakarta, hlm. 6.

a. *Central Processing Unit (CPU)*

CPU berfungsi untuk mengambil instruksi dari memori, mendekodekannya dan kemudian mengeksekusi instruksi tersebut. Selama proses tersebut CPU akan menghasilkan sinyal kontrol, memindahkan data ke *I/O port* atau sebaliknya, melakukan fungsi aritmatika dan logika juga mendeteksi sinyal dari luar CPU. CPU bertugas menghubungkan peralatan *input* dan *output*. CPU juga menangani masalah komunikasi data dengan piranti *eksternal*, interkoneksi antar bagian-bagian *internal* PLC, eksekusi program, manajemen memori, mengawasi atau mengamati masukan dan memberikan sinyal ke *output*. Kontroler PLC ini memiliki kemampuan untuk memeriksa memori secara teratur dan kompleks untuk mengetahui rusak atau tidaknya memori pada PLC. Untuk mengetahui kesalahan atau kerusakan pada PLC dapat dilihat pada indikator *LED* PLC.

b. Memori

Memori sistem ini berfungsi untuk menyimpan sistem operasi dan juga untuk menyimpan program yang harus dijalankan dalam bentuk *biner*, hasil terjemahan *ladder diagram* yang dibuat oleh pemrogram. Data-data yang terdapat pada memori dapat diubah (dikosongkan atau dihapus). Pemrograman PLC biasanya dilakukan melalui kanal serial komputer yang dipakai untuk

memprogram. Memori pengguna terdiri dari beberapa blok yang memiliki fungsi khusus. Beberapa bagian pada memori digunakan untuk menyimpan status *input* dan *output*. Status yang sesungguhnya dari *input* maupun *output* akan disimpan sebagai *logic*/bilangan 0 dan 1. Masing-masing *input* berkaitan dengan *bit* yang terdapat dalam memori. Bagian lain dalam memori digunakan untuk menyimpan variabel yang telah dituliskan di dalam program.

PLC menggunakan peralatan memori semi konduktor seperti RAM (*Random Acces Memory*), ROM (*Read Only Memory*), dan PROM (*Programmable Read Only Memory*). RAM mempunyai waktu akses yang cepat dan program-program yang terdapat didalamnya dapat diprogram ulang sesuai dengan keinginan pemakaiannya. RAM disebut juga sebagai *volatile memory*, maksudnya program-program yang terdapat mudah hilang jika *supply* listrik padam. Dengan demikian untuk mengatasi *supply* listrik yang padam tersebut maka diberi *supply* cadangan daya listrik berupa baterai yang disimpan pada RAM. Seringkali CMOS RAM dipilih untuk pemakaian *power* yang rendah. Baterai ini mempunyai jangka waktu kira-kira lima tahun sebelum harus diganti.



c. Pemrograman PLC

*Controller* PLC dapat diprogram melalui komputer, namun ada juga cara lain untuk memprogramnya yaitu dengan cara memprogram manual atau yang biasa disebut dengan konsol (*console*). Untuk melakukan pemrograman manual ini dibutuhkan *software* yang sesuai dengan produk PLC nya, karena masing-masing produk PLC membutuhkan *software*-nya sendiri-sendiri. Dalam memprogram ulang PLC sangat penting diperhatikan bahwa saat sistem diperbaiki program yang benar dan sesuai harus disimpan kedalam PLC lagi. Penting dilakukan pemeriksaan program PLC apakah selama penyimpanan tidak terjadi perubahan atau sebaliknya, apakah program sudah berjalan benar atau tidak. Karena dengan pemeriksaan ini dapat menghindari situasi berbahaya dalam ruang produksi (pabrik).

d. *Power Supply* PLC

*Power supply* digunakan untuk memberikan aliran daya ke seluruh bagian PLC termasuk CPU, Memori, dan lain-lain. Rata-rata PLC bekerja dengan daya 24 VDC atau 220 VAC. Untuk PLC besar catu daya biasanya terpisah sebagai modul sendiri sedangkan untuk PLC kecil catu dayanya sudah menyatu dengan PLC nya.

e. *Input PLC*

Kecerdasan suatu sistem yang terotomasi sangat bergantung pada kemampuan sebuah PLC untuk membaca sinyal dari berbagai macam jenis sensor dan piranti *input* yang lain. Untuk mendeteksi proses yang sedang terjadi maka dibutuhkan sensor-sensor yang tepat untuk masing-masing kondisi. Sinyal-sinyal yang telah terdeteksi sensor tersebut dapat berupa *logic* (*on* atau *off*) maupun *analog*. Pada PLC kecil biasanya hanya mampu menerima *inputdigital* saja namun untuk PLC besar mampu menerima *inputanalog* dari *unit* khusus yang sesuai dengan PLC nya.

Setiap *input* memiliki alamat dan nomor urutan khusus yang digunakan selama membuat program untuk memonitor satu persatu aktivitas *input* didalam program. Indikasi urutan status dari *input* ditandai *light emitting diode* (LED) pada PLC atau modul *input*, hal ini dimaksudkan untuk memudahkan pengecekan proses pengoperasian *input* dari PLC itu sendiri.

f. *Interfacing Input*

*Interfacinginput* berada dijalur *input* dan CPU. Tujuan elemen ini adalah untuk melindungi CPU dari sinyal-sinyal yang tidak diinginkan dan dapat merusak CPU tersebut. Modul *interfacing input* ini berfungsi untuk mengkonversi sinyal-sinyal *input* dari

luar ke sinyal yang sesuai dengan tegangan kerja CPU. Contoh *input* tegangan pada sensor yaitu 22 VDC maka harus dikonversikan ke 5 VDC agar sesuai dengan tegangan kerja CPU.

g. *Output PLC*

Sistem terotomasi yang memiliki *input* pastinya juga memiliki *output*. Tidak akan lengkap jika tidak ada jalur *input* yang digunakan untuk menghubungkan ke alat-alat *eksternal*. Contoh alat yang banyak digunakan yaitu motor, solenoida, relai, lampu indikator, *speaker*, dan sebagainya. *Output* ini dapat berupa *analog* maupun *digital*. *Output* digital akan dianalogikan seperti saklar yang dapat memutus atau menyambungkan jalur. Untuk *output analog* contohnya dapat digunakan merubah tegangan untuk pengendalian motor secara regulasi *linier* sehingga memperoleh kecepatan yang diinginkan.

h. *Interfacing Output*

Seperti halnya *interfacing input*, *output* juga memerlukan *interfacing* yang sama digunakan untuk memberikan perlindungan antara CPU dengan peralatan *eksternal* agar tidak terjadi kerusakan pada CPU-nya.

### i. Jalur Tambahan

Setiap PLC biasanya memiliki jumlah *input* dan *output* yang terbatas. Jika diperlukan jumlah ini dapat ditambahkan dengan menggunakan modul *input* dan *output* tambahan.

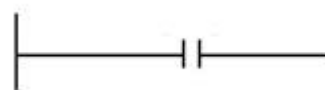
### 2.1.3.6. Instruksi – Instruksi Dalam Pemograman PLC

#### 1. Instruksi Dasar

Instruksi dasar merupakan instruksi yang digunakan untuk membuat rangkaian logika dari diagram tangga pada pemrograman PLC.<sup>7</sup> Instruksi dasar pada pemograman diagram tangga ini ada enam, yaitu :

#### a. LD

LD atau singkatan dari *Load*, merupakan instruksi untuk memulai program garis atau blok pada rangkaian logika yang dimulai dengan kontak NO (*Normally Open*) input seperti terlihat pada gambar 2.6.



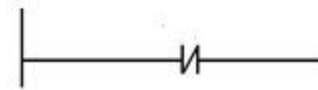
**Gambar 2.6. Instruksi LD**

Sumber: dokumentasi

<sup>7</sup>Husanto, Thomas, ST, MT, "PLC (*Programmable Logic Controller*) FP Sigma". C.V Andi, Yogyakarta, hlm.47.

b. NOT

Instruksi dasar NOT berfungsi untuk membentuk suatu kontak NC (*Normally Close*) input seperti pada gambar 2.7.



**Gambar 2.7. Instruksi NOT**

Sumber: dokumentasi

c. OUT

OUT merupakan instruksi untuk memasukkan program koil *output*. Kontak-kontak dari masing-masing koil *output* dapat digunakan beberapa kali sesuai yang diinginkan. Simbol instruksi OUT bisa dilihat pada gambar 2.8.

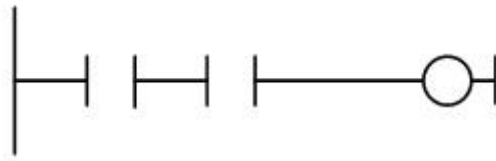


**Gambar 2.8. Instruksi OUT**

Sumber: dokumentasi

## d. AND

Instruksi AND ini digunakan untuk menghubungkan dua atau lebih kontak-kontak *input* atau *output* secara seri seperti pada gambar 2.9.

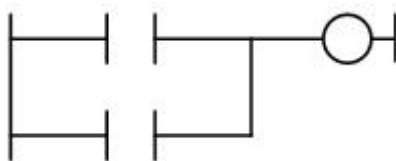


**Gambar 2.9. Instruksi AND**

Sumber: dokumentasi

## e. OR

Instruksi dasar OR digunakan untuk menghubungkan dua atau lebih kontak-kontak *input* atau *output* secara parallel seperti terlihat pada gambar 2.10.



**Gambar 2.10. Instruksi OR**

Sumber: dokumentasi

f. END

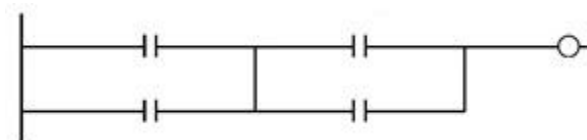
Instruksi dasar END untuk menyatakan rangkaian kontrol yang dibuat telah berakhir. Instruksi END ini harus selalu dimasukkan dalam penulisan program, karena apabila akhir rangkaian kontrol tidak dilengkapi dengan instruksi END, maka program tersebut tidak akan dieksekusi oleh CPU.

2. Instruksi Gabungan

Instruksi gabungan merupakan suatu instruksi yang menggunakan dua buah instruksi dasar atau lebih untuk menggabungkan dua blok rangkaian dalam program.<sup>8</sup> Instruksi gabungan tersebut adalah sebagai berikut:

a. AND LD

Instruksi ini merupakan gabungan dari instruksi AND dan LD yang digunakan untuk menggabungkan dua blok rangkaian dalam secara seri seperti terlihat pada gambar 2.11.



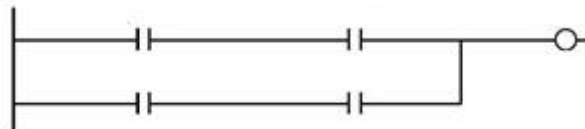
**Gambar 2.11. Instruksi AND LD**

Sumber: dokumentasi

<sup>8</sup>Syufrijal, S.T, M.T. "PLC (Konsep, aplikasi dan komunikasi jaringan PLC)". Jakarta, 2008, hlm. 6.

b. OR LD

Instruksi ini digunakan untuk menggabungkan dua blok dalam rangkaian secara parallel seperti terlihat pada gambar 2.12.



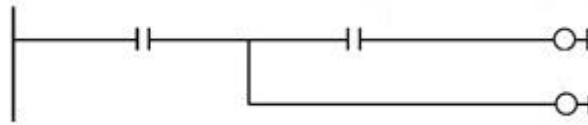
**Gambar 2.12. Instruksi OR LD**

Sumber: dokumentasi

c. Instruksi Garis Bercabang (*Temporary Relay*)

Instruksi garis bercabang merupakan suatu instruksi yang mempunyai sebuah garis yang terdiri dari dua instruksi atau lebih dan letaknya setelah *input*. Instruksi garis bercabang tersebut terdapat pada *temporary relay* (TR). *Temporary relay* adalah relay bantu yang digunakan pada rangkaian yang mempunyai dua atau lebih percabangan dari relay *output*, *timer* atau *counter*. Beberapa TR dapat digunakan sesuai dengan kebutuhan, akan tetapi tidak dapat melebihi dari jumlah yang ditetapkan pabrik. Dalam rangkaian, satu nomer TR hanya dapat digunakan sekali saja, sedangkan untuk pengalamatan TR lainnya digunakan pada rangkaian lainnya secara berurutan. Penggunaan TR dapat dilihat pada gambar 2.13.





**Gambar 2.13. Instruksi Garis Bercabang**

Sumber: dokumentasi

### 3. Instruksi *Timer* dan *Counter*

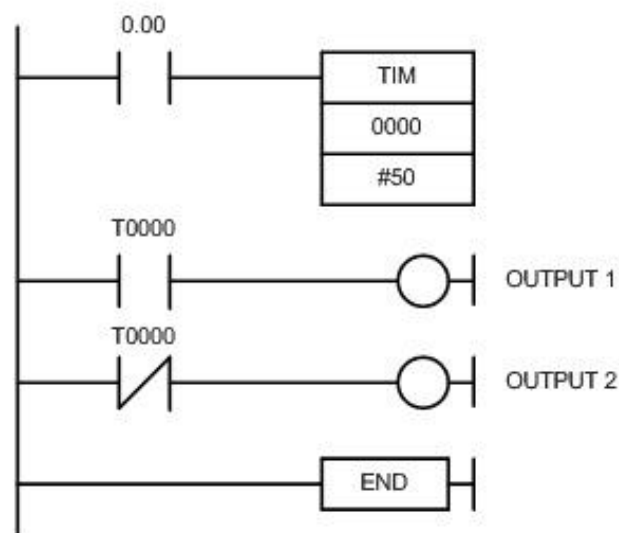
*Timer* PLC yang banyak digunakan dalam industri adalah *timer on-delay*. Keuntungan *timer* PLC adalah akurasi waktunya tinggi karena PLC menggunakan teknologi *solid-state*. Pada PLC Omron pengalamatan *timer* dan *counter* digunakan secara bersama-sama yaitu dari nomer 0 – 127. Cara kerja *timer* dan *counter* hampir mirip, perbedaannya *timer* mencacah pulsa *internal* sedangkan *counter* mencacah pulsa dari luar.

#### a. *Timer*

*Timer* adalah suatu instruksi yang membuat suatu proses berhenti sesaat sebelum kembali melanjutkan proses.<sup>9</sup> *Timer* berfungsi untuk mengaktifkan suatu keluaran dengan interval waktu yang dapat diatur. Pengaturan waktu dilakukan melalui nilai *setting* (*preset value*). *Timer* pada PLC Omron diberi nomer dari 000 – 127 (T0 – T127). Instruksi *timer* ada 2 macam yaitu *timer* (TIM) dan *high timer* (TIMH). Bedanya pada pengukuran waktu TIM

<sup>9</sup>Husanto, Thomas, ST, MT, "PLC (*Programmable Logic Controller*) FP Sigma". C.V Andi, Yogyakarta, hlm.63.

mempunyai pulsa *clock* lebih panjang dibanding TIMH. TIM mempunyai pulsa *clock* sebesar 0,1 detik sedangkan TIMH sebesar 0,01 detik. *Timer* tersebut akan bekerja bila diberi *input* dan mendapat pulsa *clock*. Untuk pulsa *clock* sudah disediakan oleh pembuat PLC. Penggunaan *timer* dapat dilihat pada gambar 2.14.



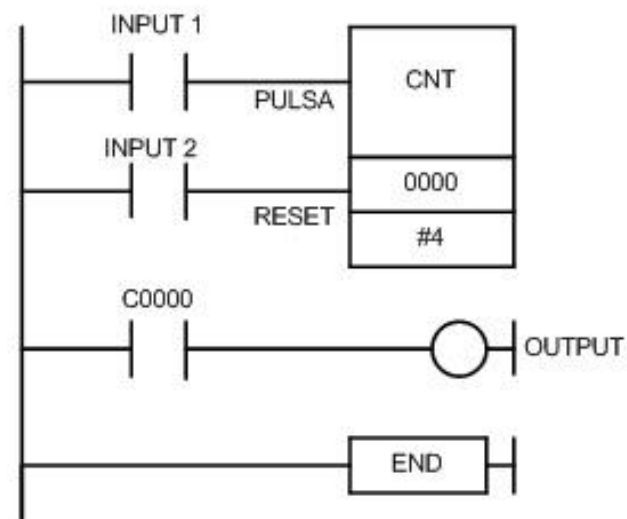
**Gambar 2.14. Contoh Penggunaan *Timer***

Sumber: dokumentasi

Saat *input* 0.00 *off*, maka *output* 1 akan *off* dan *output* 2 *on*, tetapi pada saat *input* 0.00 *on* maka *timer* mulai mencacah pulsa dari 0 sampai *preset value* (selama 5 detik) maka akan mengaktifkan *output* 1 dan mematikan *output* 2. Akan tetapi apabila *input* *off* sebelum *timer* mencapai *preset value* maka *timer* akan *off* (*reset*) sehingga menyebabkan *output* 1 *off* dan *output* 2 *on* kembali.

*b. Counter*

Instruksi *counter* pada PLC berfungsi untuk menghitung setiap *input* yang masuk.<sup>10</sup> Pada PLC Omron terdapat *counter* yang diberi nomor dari 0 – 127 (C0 – C127). Penggunaan alamat *counter* ini digunakan bersama-sama dengan *timer*. Oleh sebab itu, dalam satu program, pemberian nomor *counter* tidak boleh sama dengan nomor *timer*. Cara kerja *counter* mirip dengan *timer*, perbedaannya *timer* mencacah pulsa *internal* sedangkan *counter* mencacah pulsa dari luar. Ada 2 sinyal *input* yang digunakan oleh *counter* yaitu sinyal pulsa dan sinyal *reset*. Penggunaan *counter* dapat dilihat pada gambar 2.15.



**Gambar 2.15. Contoh Penggunaan Counter**

Sumber: dokumentasi

<sup>10</sup>Husanto, Thomas, ST, MT, "PLC (Programmable Logic Controller) FP Sigma". C.V Andi, Yogyakarta,,hlm.68.

*Counter* akan mulai mencacah pulsa dari 0 sampai *preset value* ketika terdapat sinyal *input 1* berupa pulsa dan kondisi *input2* sebagai resetya *off*. Bila cacahan *counter* sudah mencapai *preset value* yaitu sebanyak 4 kali maka *counter* akan mengaktifkan *output*. Akan tetapi bila *input2* reseton sebelum *counter* mencapai *preset value* maka *counter* akan *off (reset)* dan *output* akan *off*.

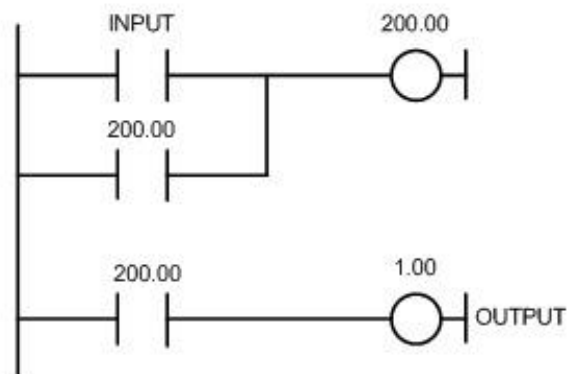
#### 4. *Internal Relay*

*Internal relay* adalah *general purpose relay* yang ada di dalam PLC yang tidak dapat diakses secara langsung untuk digunakan sebagai *input* maupun *output* seperti yang terdapat pada program komponen. *Internalrelay* adalah *relay* semu yang merupakan *bit* digital yang disimpan pada *internal image register*. Dari sudut pandang pemrograman, semua PLC mempunyai satu *coil* dan mempunyai sejumlah kontak N/O dan N/C sesuai dengan yang diinginkan *programmer*, contoh penggunaan internal relay bisa dilihat pada gambar 2.16. Semua PLC mempunyai *internal relay* akan tetapi penomeran dan jumlah maksimum yang diperbolehkan tergantung dari merek dan model PLC.

*Internalrelay* memberi keleluasaan pada *programmer* untuk melaksanakan operasi *internal* yang lebih rumit tanpa memerlukan penggunaan biaya mahal untuk beberapa *output relay*. Dalam contoh

pemrograman pada PLC Omron memakai simbol IR dengan penomoran sebagai berikut :

- a. 200 – 231
- b. 600 – 615
- c. 700 – 715
- d. 800 – 815
- e. 900 – 915



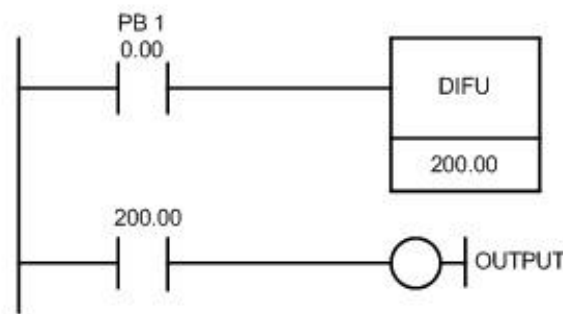
**Gambar 2.16.***Internal Relay*

Sumber: dokumentasi

### 5. DIFU (*Differential Up*)

Aplikasi kontrol ini berfungsi untuk mengon kan *output* selama satu *scan*.<sup>11</sup> Aplikasi DIFU bisa dilihat pada gambar 2.17. Contoh program menggunakan DIFU :

Apabila PB 1 ditekan maka *output* DIFU 200.00 akan *on* dan kontak DIFU 200.00 akan hidup hanya sekejap walaupun tombol PB 1 ditekan lama.



**Gambar 2.17. Contoh Penggunaan DIFU**

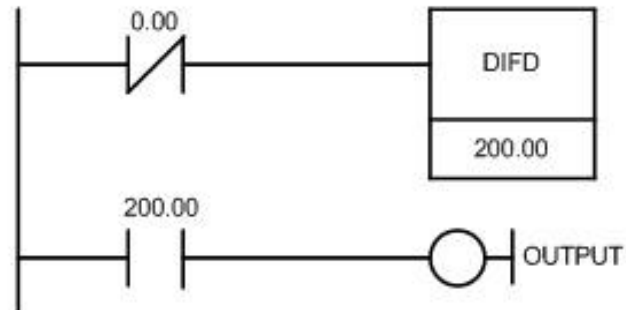
Sumber: dokumentasi

### 6. DIFD (*Differential Down*)

Aplikasi kontrol ini berfungsi untuk melewatkan satu pulsa *input* (*On – Off*) pada *output* sehingga *output* hanya akan *on* selama satu pulsa (satu *scan*) seperti terlihat pada gambar 2.18. Contoh program menggunakan DIFD :

*Output* DIFD 200.00 akan *on* selama satu *scan* setelah *input* 0.00 *off*.

<sup>11</sup>Husanto, Thomas, ST, MT, "PLC (*Programmable Logic Controller*) FP Sigma". C.V Andi, Yogyakarta,,hlm.59.



**Gambar 2.18. Contoh Penggunaan DIFD**

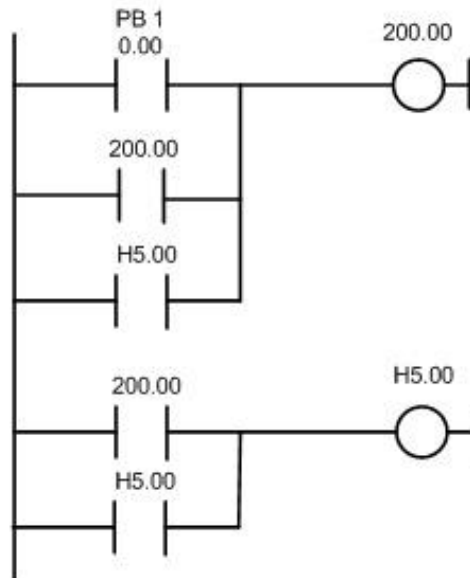
Sumber: dokumentasi

### 7. *Holding Relay*

*Holding relay* adalah  *relay internal* yang bisa dipakai untuk menahan sistem yang sedang bekerja walaupun aliran  *supply power off* seperti yang terlihat pada gambar 2.19, misalnya jika sumber  *power/PLN* mati, apabila pada program dipasang  *holding relay* maka proses bisa tetap lanjut tanpa harus memulai program dari awal lagi.

Contoh penggunaan  *holding relay* :

Apabila  *input* 0.00 PB 1 dihidupkan maka  *output* 200.00 dan H5.00 akan hidup, dan apabila sumber  *power/PLN* mati maka  *output* 200.00 masih tetap  *ON* karena program ditahan oleh  *Holding relay*.



**Gambar 2.19. Contoh Penggunaan *Holding Relay***

Sumber: dokumentasi

#### 2.1.3.7. *Software CX-Programmer*

*CX-Programmer* merupakan salah satu bentuk perangkat lunak yang digunakan untuk memasukkan program ke dalam PLC. Berikut ini adalah langkah-langkah yang diperlukan dalam membuat program PLC melalui *CX-Programmer*, yaitu:

1. *Install Software CX-Programmer*

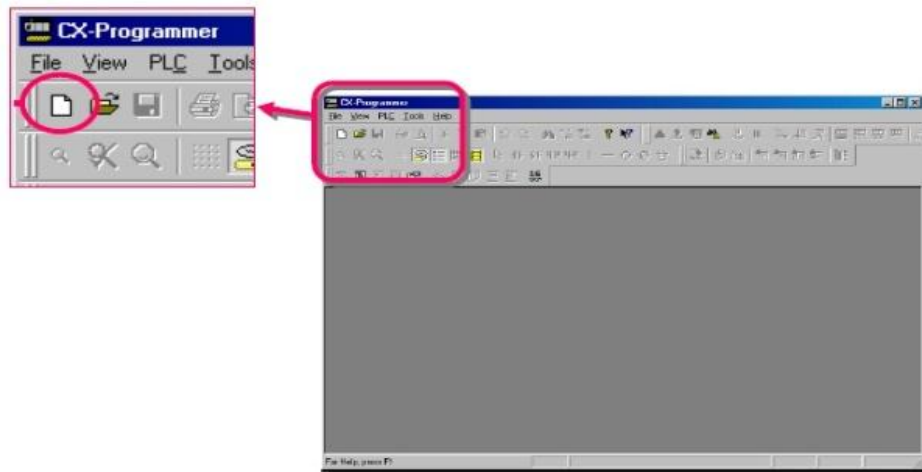
Dalam penginstalan *CX-Programmer* perlu dipastikan untuk menutup semua *windows* program yang sedang aktif. Jika kita memiliki program *CX-Programmer* versi lama, *uninstal* terlebih dahulu sebelum meng-*install CX-Programmer* versi terbaru. *CX-*



*Programmer* dapat di-*install* mulai dari OS Windows 95/98/NT 4.0 SP 6, Windows 2000/Me, hingga Windows XP dan 7.

## 2. Membuat *Project* Baru

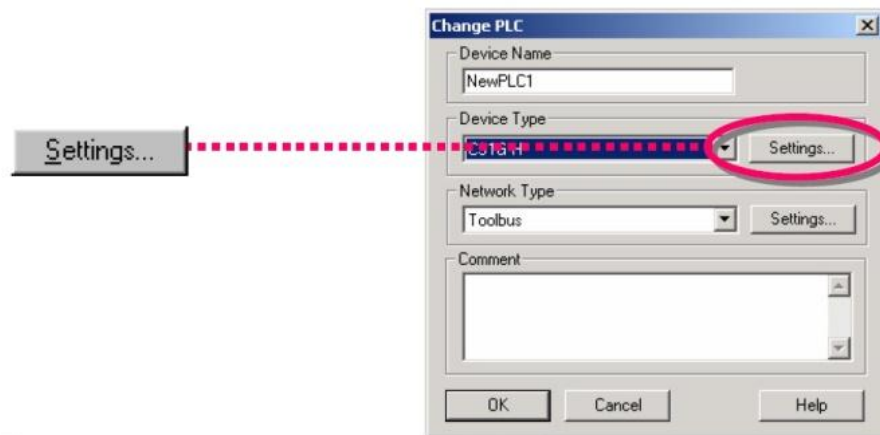
Untuk membuat *project* baru, langkah yang harus dilakukan pertama kali adalah dengan mengklik [*New*] pada *toolbar* di *CX-Programmer*, seperti pada gambar 2.20.



**Gambar 2.20. Tombol *New* pada *Toolbar***

Sumber: dokumentasi

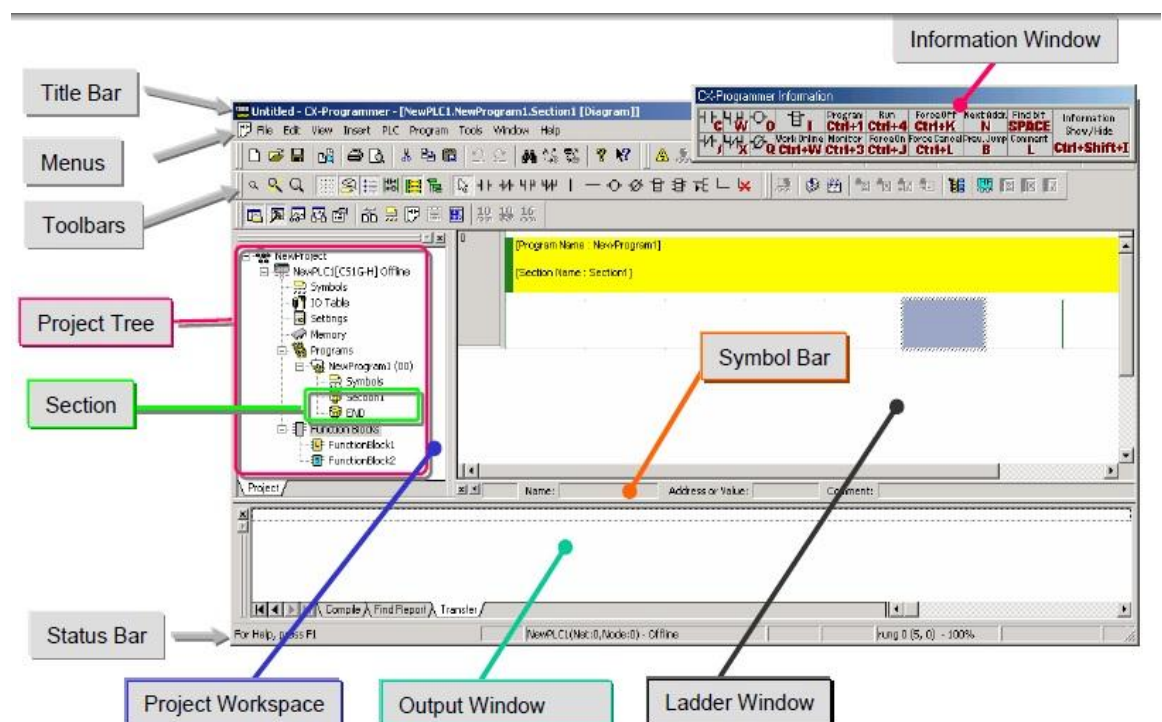
Selanjutnya akan muncul tampilan seperti gambar 2.21 untuk memilih jenis PLC yang akan digunakan, kemudian klik tombol *settings* untuk menampilkan layar (*Device Type Settings*) dan mengubah jenis CPU PLC.



**Gambar 2.21. Change PLC**

Sumber: dokumentasi

Kemudian klik tombol OK. Maka akan muncul *main window* seperti pada gambar 2.22.



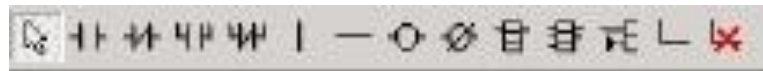
**Gambar 2.22. Main Window Software CX-Programmer**

Sumber: dokumentasi

### 3. Membuat Program

Untuk membuat program dengan *ladder diagram*, dapat mengklik simbol kontak, koil, garis atau fungsi yang diinginkan seperti yang terdapat pada gambar 2.23. untuk membuat suatu program pada PLC, dilakukan langkah-langkah pemrograman sebagai berikut :

- a. Tentukan sistem apa yang akan dikontrol.
- b. Hitung jumlah *input/output* dan diberi alamatnya.
- c. Membuat *ladder diagram*.
- d. *Test* dan *check* program.



**Gambar 2.23. Simbol Pada Toolbar**

Sumber: dokumentasi

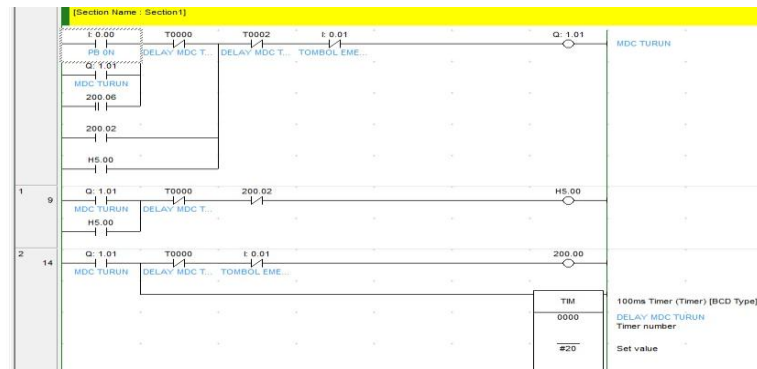
Fungsi masing-masing menu pada gambar 2.27 dijelaskan pada tabel

2.1:

**Tabel 2.1. Fungsi Main Window Software CX-Programmer**

Nama	Fungsi
<i>Title Bar</i>	Memperlihatkan nama <i>file</i> yang telah di <i>save</i> pada <i>CX-Programmer</i>
<i>Menu</i>	Untuk memilih <i>item</i> menu
<i>Toolbars</i>	Untuk memilih fungsi yang akan digunakan. Pilih [View] → [Toolbars], untuk memperlihatkan <i>toolbars</i> .
<i>Section</i>	Untuk membagi suatu program dalam beberapa <i>block</i>
<i>Project Workspace</i> <i>Project Tree</i>	Mengontrol program dan data. Dapat digunakan untuk meng- <i>copy</i> data dengan <i>Drag and Drop</i> di antara <i>project</i> yang berbeda atau dalam satu <i>project</i> .
<i>Ladder Window</i>	Layar untuk menulis dan mengedit <i>diagram ladder</i>
<i>Output Window</i>	<ul style="list-style-type: none"> <li>• Menunjukkan <i>error check</i></li> <li>• Menunjukkan hasil pencarian <i>contacts/coils</i> di <i>list form</i></li> <li>• Menunjukkan <i>error details</i> ketika terjadi kesalahan dalam suatu <i>file project</i>.</li> </ul>
<i>Status Bar</i>	Menunjukkan informasi seperti nama PLC, <i>online/offline</i> , lokasi <i>cell</i> yang aktif.
<i>Information Window</i>	Layar <i>small window</i> untuk menunjukkan <i>basic shortcut keys</i> yang digunakan di <i>CX-Programmer</i> . Munculkan pilih [View] -> [Information Window].
<i>Symbol Bar</i>	Menunjukkan nama, alamat atau nilai, dan penjelasan dari simbol yang dipilih kursor.

Lalu buat *ladder diagram* seperti yang terlihat pada gambar 2.24 sesuai logika yang diinginkan untuk ditransfer ke PLC kemudian dijalankan pada alat yang terhubung dengan PLC.<sup>12</sup>



**Gambar 2.24. Membuat Program yang Diinginkan**

Sumber: dokumentasi

4. Aturan-aturan pada Pemrograman *Ladder Diagram*
  - a. *Output* dapat menjadi *input*, tetapi *input* tidak dapat menjadi *output*.

*Output* PLC dapat berubah menjadi *input*, di mana *input* tersebut baru akan aktif jika *output* diaktifkan. Hal ini dimungkinkan karena *output* tersebut merupakan bagian alamat dari PLC. Jadi yang dimanipulasi ialah alamat *output*, bukan peralatan *output* secara fisik.

Namun hal tersebut tidak berlaku untuk *input*, di mana *input* tidak dapat berubah menjadi *output*. Alasannya

<sup>12</sup>Syufrijal, S.T, M.T. "PLC (Konsep, aplikasi dan komunikasi jaringan PLC)". Jakarta, 2008, hlm.47.

sederhana saja, untuk mengaktifkan *input*, diperlukan suatu tindakan atau perubahan fisik. Misalkan untuk mengaktifkan tombol harus dilakukan penekanan tombol secara fisik. Hal tersebut tidak bisa dilakukan dari program PLC (meskipun ada fitur *force* pada PLC yang memungkinkan pengecekan *input*, namun fitur tersebut tidak ditujukan dalam konteks pemrograman.

b. *Internal relay* dapat digunakan sebagai perantara

Pada era *relay*, seluruh peralatan *input* dan *output* dihubungkan dengan *relay* sebagai pengendali. Pada PLC, sebagai gantinya diberikan *relay virtual* yang disebut *internal relay*. Perbedaan *internal relay* dengan *input* (I) atau *output* (O) ialah tidak ada keharusan menghubungkan alat fisik tertentu pada alamat ini. Sedang pada alamat *input* atau *output*, penggunaan harus benar-benar menghubungkan peralatan secara fisik.

c. *Input* dapat muncul berkali-kali, namun *output* hanya boleh muncul 1 kali.

Seperti halnya *contact* pada *relay*, kontak di PLC dapat muncul berkali-kali dalam suatu *ladder diagram*. Ini adalah salah satu kelebihan PLC dibanding *relay*, karena jumlah *contact* maksimal yang umum beredar di pasaran ialah 4

*contact* saja. Sedangkan jumlah maksimal kontak pada PLC nyaris tak terbatas (hanya dibatasi oleh ketersediaan memori PLC saja).

#### 2.1.4. Motor AC

Motor AC merupakan sebuah motor listrik yang tenaga penggerakannya berasal dari arus bolak-balik (arus AC). Motor AC lebih sering digunakan dalam industri dari pada motor DC. Tetapi motor AC ini tidak dapat beroperasi dengan baik pada kecepatan rendah. Motor AC biasanya terdiri dari dua bagian dasar<sup>13</sup>, yaitu:

##### a. *Stator*

Stator merupakan bagian dari motor yang tidak bergerak (stasioner/statis). Stator berupa kumparan yang dialiri dengan arus bolak-balik untuk menghasilkan medan magnet yang berputar. *Stator* ini terbentuk atas lapisan plat-plat tipis dengan sejumlah *pole* yang tersusun melingkar, seperti jari-jari pada roda. Seutas kawat tembaga dililitkan sebanyak sekian lilitan/putaran di tiap-tiap *pole*. Bentuk Stator dapat dilihat pada gambar 2.25.

---

<sup>13</sup>“Jenis-jenis Motor AC” <http://blogs.itb.ac.id/el2244k0112211077alpinarief/2013/05/02/motor-ac/>, terakhir diakses 8 Agustus 2015, jam 20.28 WIB.



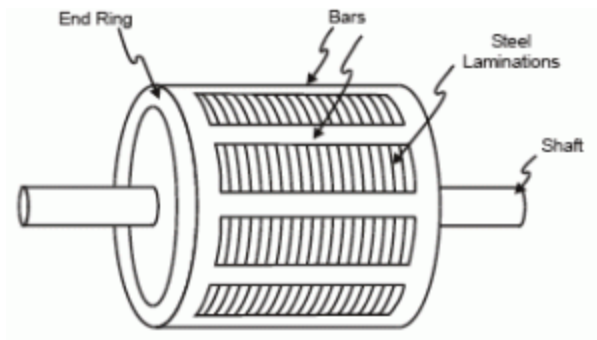
**Gambar 2.25. Bentuk *Stator***

b. *Rotor*

Rotor merupakan bagian dari motor listrik yang mengalami perputaran. Perputaran rotor disebabkan karena adanya medan magnet dan lilitan kawat pada rotor. Sedangkan torsi dari perputaran rotor ditentukan oleh banyaknya lilitan kawat dan juga diameternya. Pada rotor terdapat kutub-kutub magnet dengan lilitan-lilitan kawatnya dialiri oleh arus searah. Kutub magnet rotor terdiri dua jenis yaitu :

1. Rotor kutub menonjol (*salient*), adalah tipe yang dipakai untuk generator-generator kecepatan rendah dan menengah.
2. Rotor kutub tidak menonjol atau rotor silinder digunakan untuk generator-generator turbo atau generator kecepatan tinggi.





**Gambar 2.26. Bentuk Rotor**

Terdapat dua jenis motor AC, tergantung pada tipe rotor yang digunakan:

- a. Tipe pertama adalah **motor induksi** atau motor asinkron. Medan magnet pada rotor motor ini diciptakan oleh arus induksi.
- b. Tipe kedua adalah **motor sinkron**, yang tidak bergantung pada induksi. sebagai hasilnya, dapat memutar tepat pada frekuensi supply atau kelipatan dari frekuensi supply.

#### 2.1.4.1. Motor Induksi (*Asynchronous*)

Motor induksi merupakan motor listrik arus bolak balik (ac) yang paling luas digunakan. Penamaannya berasal dari kenyataan bahwa motor ini bekerja berdasarkan induksi medan magnet stator ke statornya, dimana arus rotor motor ini bukan diperoleh dari sumber tertentu, tetapi merupakan arus yang terinduksi sebagai akibat adanya perbedaan relatif antara putaran rotor

dengan medan putar (*rotating magnetic field*) yang dihasilkan oleh arus stator.

Motor induksi disebut sebagai motor asinkron atau rotating transformer. Jenis Motor AC ini menggunakan induksi elektromagnetik untuk membuat kumparan berputar. Pada jenis ini rotor berputar lebih lambat daripada arus yang disediakan untuk memutarinya. Induksi electromagnet ini menyebabkan medan magnet yang menyelubungi rotor motor tersebut. Motor induksi sangat banyak digunakan di dalam kehidupan sehari-hari baik di industri maupun di rumah tangga.

#### 2.1.4.1.1. Prinsip Kerja Motor Induksi

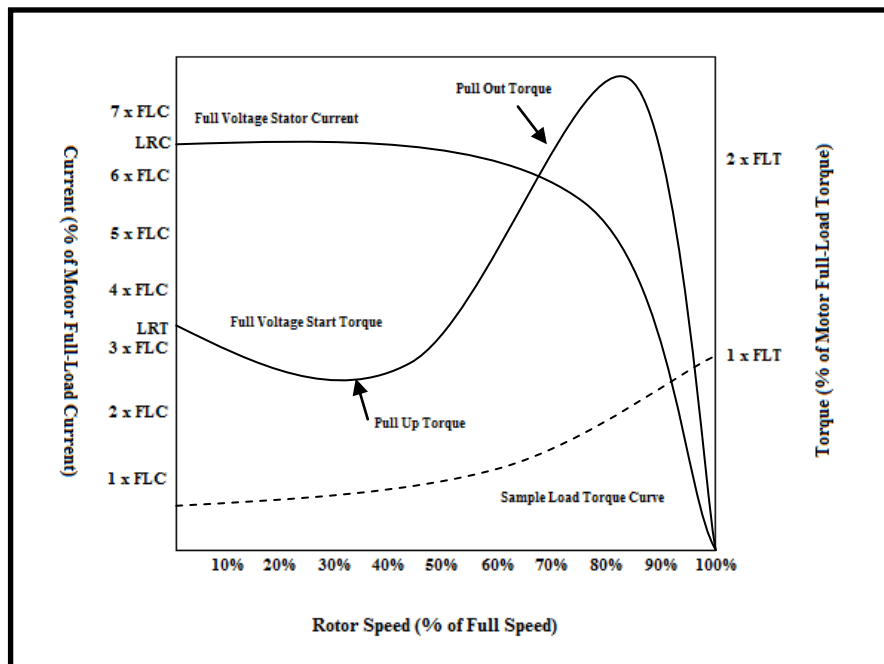
Prinsip kerja dari motor induksi yaitu suplai listrik di stator akan menghasilkan medan magnet. Medan magnet ini bergerak dengan kecepatan sinkron di sekitar rotor. Arus dari rotor menghasilkan medan magnet yang kedua dan berusaha melawan medan magnet stator sehingga menyebabkan rotor berputar. Tetapi pada kenyataannya motor tidak pernah bekerja pada kecepatan sinkron tetapi pada kecepatan dasar yang lebih rendah. Perbedaan tersebut disebabkan karena adanya slip yang meningkat dengan meningkatnya beban. Untuk menghindari slip maka dapat dipasang sebuah cincin geser atau slip ring

$$\% \text{Slip} = ((N_s - N_b) / N_s) \times 100$$

Dimana:

$N_s$  = kecepatan sinkron dalam RPM

$N_b$  = kecepatan dasar dalam RPM



**Gambar 2.27. Grafik Hubungan Antara Beban, Kecepatan , dan Torsi**

Gambar diatas menunjukkan hubungan antara torsi-kecepatan pada motor induksi 3 fasa.

Interpretasi grafik:

- Pada kondisi baru mulai menyala terdapat arus dengan nilai awal yang besar dan torsi yang rendah.

- b. Mencapai 80% kecepatan maksimum, torsi berada pada tingkat tertinggi (pull out torque)
- c. Pada kecepatan penuh atau kecepatan sinkron, arus torsi dan stator turun ke nol.

#### 2.1.4.2. Motor Sinkron (*synchronous*)

Synchronous motor adalah motor AC tiga-fasa yang dijalankan pada kecepatan sinkron, tanpa slip. Synchronous motor adalah motor AC tiga-fasa yang dijalankan pada kecepatan sinkron, tanpa slip. Motor sinkron merupakan motor arus bolak-balik ( AC ) yang penggunaannya tidak seluas motor asinkron. Secara umum penggunaan motor sinkron difungsikan sebagai generator, akan tetapi motor sinkron tetap digunakan oleh industri yang membutuhkan ketelitian putaran dan putaran konstan.

Synchronous Motor biasanya sebuah motor AC yang memiliki rotor berputar pada kecepatan yang sama seperti arus bolak-balik yang sedang disediakan untuk memutar kumparannya. *Slip rings* atau magnet permanen yang dikenakan menghasilkan medan magnet di sekitar rotor.

Sebuah motor sinkron selalu beroperasi pada kecepatan konstan, pada kondisi tidak berbeban. Tetapi apabila motor diberi beban, maka motor akan selalu akan berusaha untuk tetap pada putaran konstan. Dan motor akan melepaskan kondisi sinkronnya apabila beban yang ditanggung terlalu besar ( Torsi Pull-out ).

Synchronous motor memiliki karakteristik sebagai berikut:

- a. Sebuah stator tiga fasa sama dengan motor induksi. Stator yang memiliki tegangan menengah sering digunakan.
- b. Sebuah rotor yang bersinggungan (bidang yang berputar) yang memiliki jumlah kutub yang sama sebagai statornya, dan dipasang oleh sumber eksternal arus DC. Tipe brush dan brushless exciter digunakan untuk memasok medan arus DC ke rotor. Arus pada rotor membentuk suatu hubungan kutub magnetik Utara-Selatan pada kutub-kutub rotor, yang memungkinkan rotor untuk “mengunci” dengan fluks stator yang berputar.
- c. Dimulai sebagai sebuah motor induksi. Rotor synchronous motor juga mempunyai sebuah squirrel-cage winding yang dikenal sebagai *Amortisseur winding*, yang berfungsi menghasilkan torsi untuk menyalakan motor.

#### 2.1.4.2.1. Komponen motor sinkron

Motor sinkron juga memiliki dua bagian penting yaitu bagian stator yang merupakan bagian komponen diam, dan bagian rotor yang berfungsi sebagai komponen berputar, stator terdiri dari inti besi dari bahan ferromagnet yang dililit dengan lilitan 3 fasa, lilitan 3 fasa ini sama dengan lilitan 3 fasa pada rotor induksi.

#### 2.1.4.2.2. Prinsip kerja motor sinkron

Berbeda dengan motor induksi, motor sinkron mendapat eksitasi dari sumber DC eksternal yang dihubungkan ke rangkaian rotor melalui slip ring dan sikat. Arus DC pada rotor ini menghasilkan medan magnet rotor ( $B_R$ ) yang tetap. Kutub medan rotor mendapat tarikan dari kutub medan putar stator hingga turut berputar dengan kecepatan yang sama (sinkron). Torsi yang dihasilkan motor sinkron merupakan fungsi sudut torsi ( $\delta$ ). Semakin besar sudut antara kedua medan magnet, maka torsi yang dihasilkan akan semakin besar seperti persamaan di bawah ini.

$$\mathbf{T} = \mathbf{k} \cdot \mathbf{B}_R \cdot \mathbf{B}_{net} \sin \delta$$

Pada beban nol, sumbu kutub medan putar berimpit dengan sumbu kumparan medan ( $\delta = 0$ ). Setiap penambahan beban membuat medan motor “tertinggal” dari medan stator, berbentuk sudut kopel ( $\delta$ ); untuk kemudian berputar dengan kecepatan yang sama lagi. Beban maksimum tercapai ketika  $\delta = 90^\circ$ . Penambahan beban lebih lanjut mengakibatkan hilangnya kekuatan torsi dan motor disebut kehilangan sinkronisasi. Oleh karena pada motor sinkron terdapat dua sumber pembangkit fluks yaitu arus bolak-balik (AC) pada stator dan arus searah (DC) pada rotor, maka ketika arus medan pada rotor cukup untuk membangkitkan fluks (ggm) yang diperlukan motor, maka stator tidak perlu memberikan arus magnetisasi atau daya reaktif dan motor

bekerja pada faktor daya = 1,0. Ketika arus medan pada rotor kurang (penguat bekurang), stator akan menarik arus magnetisasi dari jala-jala, sehingga motor bekerja pada faktor daya terbelakang (*lagging*). Sebaliknya bila arus pada medan rotor berlebih (penguat berlebih), kelebihan fluks (ggm) ini harus diimbangi, dan stator akan menarik arus yang bersifat kapasitif dari jala-jala, dan karenanya motor bekerja pada faktor daya mendahului (*leading*). Dengan demikian, faktor daya motor sinkron dapat diatur dengan mengubah-ubah harga arus medan (IF).<sup>14</sup>

#### 2.1.5. Planetary Gear

*Gear* merupakan element umum dalam dunia *mechanical* dan peranan pentingnya semakin meningkat seiring dengan proses perkembangan teknologi pada mesin. Pada awalnya karena belum ditemukan material dan teknik perancangan yang sesuai *gear* merupakan komponen yang tidak praktis, berisik dan tidak efisien. Tapi untuk saat ini kualitas dari *gear* baik dari material maupun *lubricant* sudah semakin baik, sehingga *gear* dapat memenuhi kebutuhan pasar, yaitu dalam penerimaan beban dan kecepatan yang lebih baik. secara umum fungsi *gear* adalah sebagai penyalur tenaga antara *shaft* yang berputar tidak pada satu sumbu dan memiliki perbedaan kecepatan. Penyalur tenaga yang lain selain gear juga dapat dipergunakan,

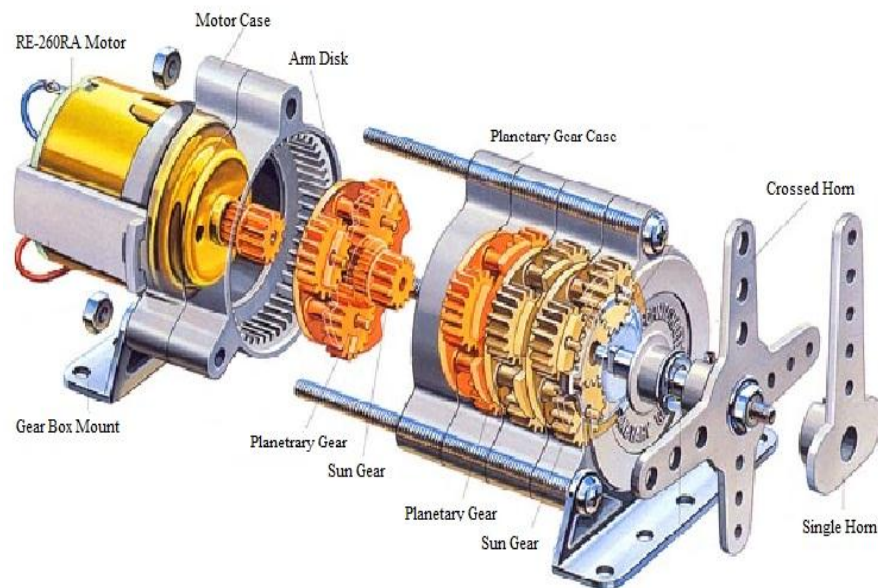
---

<sup>14</sup>Faizal Nizbah, Prinsip Kerja Motor.

<http://faizalnizbah.blogspot.co.id/2013/08/pengertian-dan-prinsip-kerja-motor.html>.  
terakhir diakses 10 Agustus 2015, jam 19.20 WIB..

namun jika dibutuhkan sebuah penggerak yang ringkas dan mengutamakan ketepatan titik “*timing*” pada *shaft*, *gear* biasanya lebih unggul dari penyalur tenaga yang lain disamping kesederhanaan, ketahanan dan efisiensinya<sup>15</sup>.

*Planetary gear* sesuai dengan namanya seperti pada gambar 2.28, merupakan susunan *gear* layaknya sebuah planet-planet yang mengelilingi pusat tata surya dengan jalur tertentu. *Planetary gear* set menyediakan peningkatan kecepatan, pengurangan kecepatan, perubahan arah, netral, dan *direct drive*. *Planetary gear* set juga dapat menyediakan variasi kecepatan disetiap tingkatan operasi.



**Gambar 2.28. Planetary Gear**

Sumber: [penonapillow.wordpress.com/2009/10/25/planetary-gear/](http://penonapillow.wordpress.com/2009/10/25/planetary-gear/)

<sup>15</sup>Rizal, *Planetary Gear*. [www.pennonapillow.wordpress.com/2009/10/25/planetary-gear/](http://www.pennonapillow.wordpress.com/2009/10/25/planetary-gear/), terakhir diakses 8 Agustus 2015, jam 20.30 WIB.



Gambar 2.28 merupakan visualisasi dari *planetary gear* dan sebuah *gear set*. Untuk sebuah *planetary gear set* sederhana terdiri dari :

- a. *Sun gear*
- b. *Carrier planetary pinion*
- c. *Ring gear* atau annulus

*Sun gear* terletak dipusat susunan. ini adalah *gear* terkecil dalam susunan dan terletak di tengah dan sebagai poros perputaran. *Sun gear* dapat berupa juga berupa rancangan spur atau *helical gear*. *Sun gear* bertautan dengan gigi pada *planetary pinion gear*. *Planetary pinion gear* adalah *gear* kecil yang disusun dalam kerangka yang disebut *planetary carrier*. *Planetary carrier* dapat terbuat dari besi tuang, aluminium atau pelat baja dan dirancang dengan sebuah *shaft* untuk masing-masing *planetary pinion gear*.

*Planetary pinion* berputar pada *needle bearing* yang diposisikan diantara *shaft planetary carrier* dan *planetary pinion*. Jumlah *planetary pinion* didalam sebuah *carrier* tergantung dari beban yang dipikul. Transmisi kendaraan otomatis harus mempunyai tiga *planetary pinion* dalam *planetary carrier*. *Heavy duty highway trucks* dapat mempunyai sebanyak 5 *planetary pinion* dalam *planetary carrier*. *Carrier* dan *pinion*-nya disebut sebagai satu kesatuan *unit gear*. *Planetary pinion* mengelilingi poros tengah *sun gear* dan dilingkari oleh *ring gear*. *Ring gear* bertindak seperti sebuah pengikat yang menahan keseluruhan *gear set* bersama dan memberikan kekuatan yang besar pada unit.

*Ring gear* diletakkan pada jarak terjauh dari poros pusat dan karena itu berfungsi sebagai tuas terbesar pada poros pusat. Untuk membantu mengingat rancangan *planetary gearset*, gunakan sistem tata surya sebagai contohnya. *Sun* adalah pusat tata surya dengan planet berputar disekelilingnya. Karena itu disebut *planetary gearset*. *Sun gear* memiliki jumlah gigi paling kecil, kemudian *ring gear* dan *planetary carrier* memiliki jumlah gigi paling banyak. Dengan mengabaikan jumlah *gear* pada *planetary carrier*, kita dapat memperkirakan jumlah gigi pada *planetary carrier* dengan cara menambahkan jumlah gigi pada *sun gear* dengan jumlah gigi pada *ring gear*. Apabila *sun gear* mempunyai 24 gigi dan *ring gear* mempunyai 48 gigi, kemudian abaikan jumlah *sun gear* dalam *planetary carrier*, *planetary carrier* akan mewakili 72 gigi.

#### 2.1.5.1 Prinsip Kerja *Planetary Gear*

Setiap komponen dalam *planetary gearset*, *sun gear*, *pinion gear*, dan *ring gear* dapat berputar atau ditahan. Perpidahan tenaga melalui sebuah *planetary gearset* hanya mungkin ketika satu komponen ditahan, atau jika dua komponen ditahan bersama. Salah satu dari tiga komponen yaitu *sun gear*, *carrier* atau *ring gear* dapat digunakan sebagai penggerak atau komponen *input*. Pada saat bersamaan, komponen yang lain tetap berputar dan kemudian menjadi kompoen yang ditahan atau diam. Komponen ketiga kemudian menjadi bagian yang digerakkan atau *output*. Tergantung

pada komponen yang menjadi penggerak, yang ditahan, dan yang digerakkan, peningkatan *torque* atau peningkatan kecepatan akan dihasilkan oleh *planetary gear set*. Arah *output* juga dapat dibalik melalui berbagai kombinasi.

#### 2.1.6. *Limit Switch*

*Limit Switch* merupakan jenis saklar yang dilengkapi dengan katup dan berfungsi menggantikan tombol. Prinsip kerja *limit switch* sama seperti saklar *Push ON* yaitu hanya akan menghubungkan pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutus saat katup tidak ditekan<sup>16</sup>. *Limit switch* seperti yang terlihat pada gambar 2.29 bisa disebut juga sensor pembatas, dalam artian mendeteksi gerakan dari suatu mesin sehingga bias mengontrolnya atau memberhentikan gerakan dari mesin tersebut sehingga dapat membatasi gerakan mesin dan tidak sampai melebihi jangkauan.

---

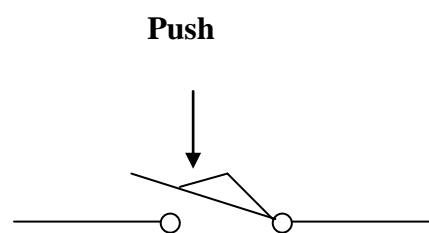
<sup>16</sup>Eldas, *Limit Switch dan Saklar Push ON*, <http://elektronika-dasar.web.id/komponen/limit-switch-dan-saklar-push-on/> terakhir diakses 8 Agustus 2015, jam 20.40 WIB.



**Gambar 2.29. Limit Switch**

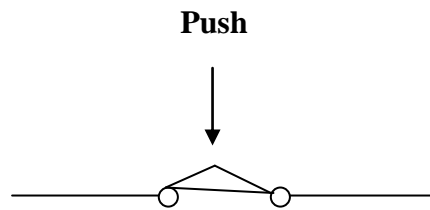
<http://elektronika-dasar.web.id/komponen/limit-switch-dan-saklar-push-on/>

Prinsip kerjanya ketika *actuator dari limit switch* tertekan suatu benda, maka *actuator* akan bergerak dan diteruskan ke bagian dalam dari *limit switch*, sehingga mengenai *micro switch* dan menghubungkan kontak-kontaknya, *pada micro switch* terdapat kontak jenis NO dan NC seperti juga sensor lainnya. Saklar pembatas kontak NO dan NC dapat dilihat pada gambar 2.30 dan 2.31.



**Gambar 2.30. Saklar Pembatas NO**

Sumber: dokumentasi



**Gambar 2.31. Saklar Pembatas NC**

Sumber: dokumentasi

Saklar pembatas dioperasikan secara mekanik, artinya saklar pembatas dikontrol oleh faktor-faktor secara otomatis, misal :tekanan dan posisi. (Gambar 2.30) menunjukkan kontak *normally open* (NO) akan membuka jika tuas kontak tidak ditekan, tetapi kontak akan menutup jika tuas kontak ditekan.(Gambar 2.31) menunjukkan kontak *normally close* (NC) akan menutup jika tuas kontak tidak ditekan, tetapi kontak akan membuka jika tuas kontak ditekan.

### 2.1.7. *Optical Proximity*



**Gambar 2.32. *Sensor Optical Proximity***

Sensor proximity merupakan suatu komponen yang berfungsi untuk mendeteksi ada atau tidaknya suatu objek. Sensor *proximity* dapat mendeteksi keberadaan benda disekitarnya tanpa ada kontak fisik dengan benda tersebut. Cara kerja sensor *proximity* ini yaitu dengan memancarkan medan elektromagnetik dan mencari perubahan bentuk medan elektromagnetik pada saat benda di deteksi. Contoh medan elektromagnetik yang sering digunakan yaitu sinar infra merah. Jika benda telah terdeteksi maka sinyal *infrared* tersebut akan merubah bentuk sinyal dan mengirimkan sinyal kembali ke sensor dan memberitahukan bahwa di depan sensor terdapat benda. Target sensor yang berbeda-beda juga membutuhkan jenis sensor *proximity* yang berbeda pula. Contohnya sensor fotolistrik kapasitif akan cocok dengan target yang mempunyai benda berbahan dasar

plastik sedangkan sensor *proximity* induktif akan mendeteksi benda berbahan dasar logam.

Jarak maksimum sensor *proximity* yang bisa terdeteksi dinamakan dengan nominal *range*. Beberapa sensor perlu diatur untuk penyesuaian nominal range nya atau dibuatkan list untuk batas kerja jaraknya.

Sensor *proximity* ini memiliki keunggulan dalam hal kemampuan yang tinggi dan umur pakai yang lama karena sensor ini tidak ada bagian mekanisnya yang kontak langsung dengan objek<sup>17</sup>

#### 2.1.7.1. Fungsi Sensor *Proximity*

- a. Mendeteksi suatu objek
- b. Mengukur dimensi suatu objek
- c. Menghitung banyaknya objek
- d. Mendeteksi symbol
- e. Pemeriksaan objek
- f. Pendeteksian warna

Selain itu ada juga fungsi yang tersirat yaitu sebagai pengontrol suatu sistem yang menggunakan sensor *proximity*. Dan juga fungsi lainnya yaitu untuk keamanan sistem.

---

<sup>17</sup>“Mengenal Sensor Proximity”, <http://www.geyosoft.com/2013/mengenal-sensor-proximity> terakhir diakses 8 Agustus 2015, jam 20.50 WIB.

### 2.1.7.2. Tipe Sensor *Proximity*

#### 1. Induktif *Proximity*

Tipe *proximity* yang bekerja berdasarkan perubahan induktansi apabila ada objek metal/logam yang berada dalam cakupan wilayah kerja sensor. Tipe ini hanya dapat mendeteksi benda logam saja dengan jarak deteksi maksimum sebesar 6 cm. Bahan dasar logam sangat mempengaruhi kemampuan pendeteksian sensor

#### 2. Kapasitif *Proximity*

Tipe *proximity* yang bekerja berdasarkan perubahan kapasitas objek yang berada pada cakupan daerah kerja sensor. Tipe ini dapat mendeteksi semua jenis benda dan memiliki jarak maksimum 2 cm.

### 2.1.7.3. Jenis Pemasangan *Proximity*

Cara pemasangan sensor *proximity* terbagi menjadi 2 yaitu :

#### 1. Flush

Flush merupakan cara pemasangan atau penanaman sensor *proximity* di dalam bahan metal. Pada tipe pemasangan ini hampir seluruh sensor ditanamkan dalam metal.

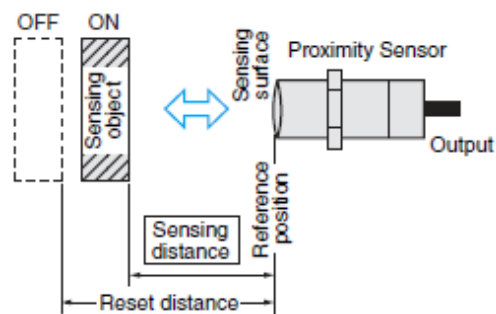


## 2. Non-Flush

Flush merupakan cara pemasangan atau penanaman sensor proximity di luar bahan metal dan diberi jarak dengan benda-benda disekitarnya.

### 2.1.7.4. Jarak dan Pengaturan Deteksi Sensor

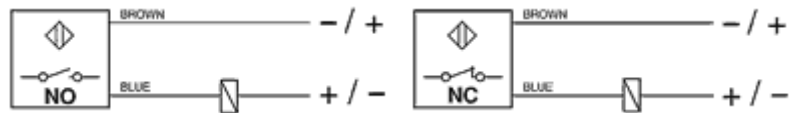
Jarak diteksi adalah jarak dari posisi yang terbaca dan tidak terbaca sensor untuk operasi kerjanya, ketika obyek benda digerakkan oleh metode tertentu.



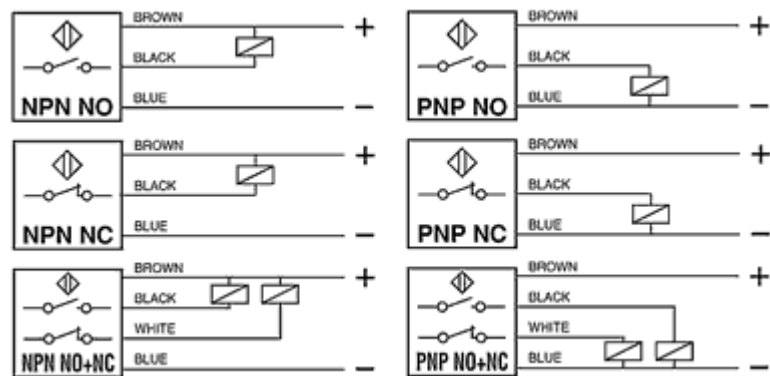
**Gambar 2.33. Jarak Deteksi Sensor *Proximity***

Mengatur jarak dari permukaan sensor memungkinkan penggunaan sensor lebih stabil dalam operasi kerjanya, termasuk pengaruh suhu dan tegangan. Posisi objek (standar)

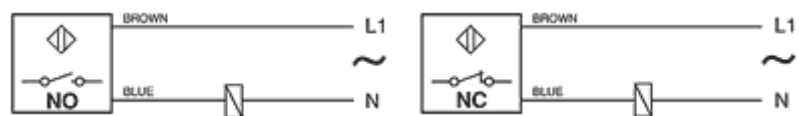
sensing transit ini adalah sekitar 70% sampai 80% dari jarak (nilai) normal sensing.<sup>18</sup>



**Gambar 2.34. Output 2 Kabel VDC**



**Gambar 2.35. Output 3 dan 4 Kabel VDC**



**Gambar 2.36. Output 2 Kabel VAC**

Dengan melihat gambar diatas kita dapat mengenali type sensor Proximity Switch ini, yaitu type NPN dan type PNP. Type inilah yang nanti bisa dikoneksikan dengan berbagai macam peralatan kontrol semi digital yang membutuhkan nilai nilai logika sebagai input untuk proses kerjanya.

<sup>18</sup>“Jarak Sensor Proximity Switch” <http://electric-mechanic.blogspot.co.id/2012/09/proximity-switch-sensor-jarak.html>. terakhir diakses 8 Agustus 2015, jam 20.55 WIB.

Beberapa jenis Proximity Switch ini hanya bisa dikoneksikan dengan perangkat PLC tergantung type dan jenisnya. Sensor ini juga bisa dikoneksikan langsung dengan berbagai macam peralatan kontrol semi digital seperti Sensor Controller dan counter relay digital.

Pada prinsipnya fungsi Proximity Switch ini dalam suatu rangkaian pengendali adalah sebagai kontrol untuk memati hidupkan suatu sistem interlock dengan bantuan peralatan semi digital untuk sistem kerja berurutan dalam rangkaian kontrol.

#### 2.1.8. Saklar *Push Button*

Saklar tekan/tombol (*push button*) seperti yang terlihat pada gambar 2.37, ada dua jenis yaitu tombol tekan *normally open* (NO) dan tombol tekan *normally close* (NC). Konstruksinya tombol tekan ada beberapa jenis, yaitu jenis tunggal *on* dan *off* dibuat secara terpisah dan ada juga yang dibuat satu tempat. Jenis ini untuk satu tombol dapat untuk *on* dan *off* tergantung keinginan penggunaannya.

Saklar *push button* berfungsi sebagai pemberi sinyal masukan pada rangkaian listrik, ketika/selama bagian knopnya ditekan maka alat ini akan bekerja sehingga kontak-kontaknya akan terhubung untuk jenis *normally open* dan akan terlepas untuk jenis *normally close*. Sebaliknya, ketika

knopnya dilepas kembali maka akan bekerja kebalikan dari sebelumnya. Tombol tekan *push button* tunggal terdiri dari dua terminal, sedangkan tombol tekan *push button* ganda terdiri dari empat terminal.<sup>19</sup>

Untuk membuktikan pada terminal *push button* bisa digunakan alat ukur tester/ohm meter, pada umumnya pemakaian terminal jenis NO digunakan untuk menghidupkan rangkaian, dan jenis NC digunakan untuk mematikan rangkaian, namun semuanya tergantung pada kebutuhan mesin.



**Gambar 2.37. Saklar *Push Button***

Sumber: <http://sugestiku.blogspot.com/2013/01/push-button-limit-switch-relay.html>

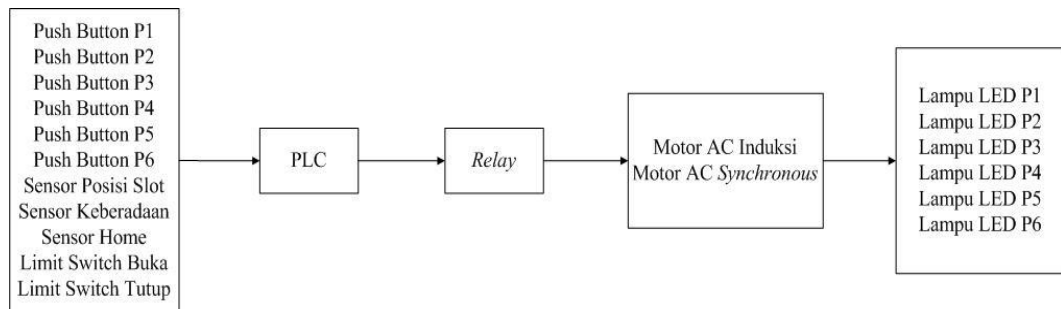
---

<sup>19</sup>Gestiyawaati, "Pushbutton, Limit Switch, Relay," Sugestiku, <http://sugestiku.blogspot.com/2013/01/push-button-limit-switch-relay.html>, terakhir diakses 15 Agustus 2015, jam 15.33 WIB.

## 2.2. Kerangka Berpikir

Penelitian Prototipe Lift ParkirKincir Otomatis Berbasis PLC didasari pada penggunaan PLC sebagai alat pengendali otomatis yang lebih handal, praktis dan aman digunakan. Prototipe berfungsi untuk menghemat penggunaan lahan dan mengefisiensikan waktu dalam memarkir sebuah kendaraan roda empat, dimana *input* yang digunakan ada 11 yaitu, *Push Button* P1 (tombol pemanggil *slot* parkir 1), *Push Button* P2 (tombol pemanggil *slot* parkir 2), *Push Button* P3 (tombol pemanggil *slot* parkir 3), *Push Button* P4 (tombol pemanggil *slot* parkir 4), *Push Button* P5 (tombol pemanggil *slot* parkir 5), *Push Button* P6 (tombol pemanggil *slot* parkir 6), dan 3 sensor yaitu sensor *Proximity Position*, sensor *Optical Proximity*, dan sensor *Proximity Home*, lalu 2 *limit switch*. *Push button* P1, P2, P3, P4, P5, dan P6 berfungsi untuk memanggil *slot* parkir mobil yang diinginkan. Sensor *Proximity Position* berfungsi untuk menghentikan posisi *slot* parkir, sensor *Optical Proximity* berfungsi untuk membaca keberadaan mobil pada *slot* parkir, dan sensor *Proximity Home* berfungsi untuk menandakan posisi *slot* parkir *Home*. Misalnya, *slot* parkir P2 ingin dipanggil ke pintu parkir, pengemudi harus menekan tombol *Push button* P2 dan sensor *Proximity Position* akan aktif untuk menghentikan *slot* parkir P2 pada pintu parkir. Kemudian *limit switch* berfungsi untuk menghentikan motor penggerak ketika pintu parkir ketika tertutup dan terbuka. Ada 8 *output* yang digunakan yaitu, Lampu *LED* P1, Lampu *LED* P2, Lampu *LED* P3, Lampu *LED* P4, Lampu *LED* P5, Lampu *LED* P6, dan 2 motor AC, yaitu motor AC induksi

dan motor AC sinkron. Lampu *LED* P1, P2, P3, P4, P5, P6 berfungsi untuk menandakan bahwa slot parkir telah terisi, lalu Motor AC induksi berfungsi sebagai motor pengendaliparkir dan motor AC sinkron sebagai motor pintu parkir. Dimana motor AC induksi berfungsi untuk menggerakkan poros *fan belt* parkir kincir dan motor AC sinkron untuk menggerakkan pintu parkir. Pengendalian otomatis yang digunakan adalah PLC CP1E. Blok diagram sistem kerja alat dapat di lihat pada gambar 2.38.



**Gambar 2.38. Blok Diagram Sistem Kerja Alat**

Sumber: dokumentasi