

BAB II

KERANGKA TEORITIS DAN KERANGKA BERPIKIR

2.1 Kerangka Teoritis

2.1.1 Sistem Kendali

Secara umum system pengendali adalah susunan komponen–komponen fisik yang di rakit sedemikian rupa sehingga mampu mengatur sistemnya sendiri atau system luarnya. Sistem control adalah proses pengaturan atau pengendalian terhadap satu atau beberapa besaran (variabel, parameter) pada suatu harga *range* tertentu. Istilah lain dari *system control* atau teknik kendali adalah teknik pengaturan, system pengendalian, atau *system* pengontrolan.

Sistem pengendalian atau teknik pengaturan juga dapat di idefinisikan suatu usaha atau perlakuan terhadap suatu *system* dengan masukan tertentu guna mendapatkan keluaran sesuai yang diinginkan.

Contoh *system* pengendalian yang paling mendasar adalah kendalion-off saklar listrik. Aktivitas menghidupkan dan mematikan saklar meyebabkan adanya situasi saklar hidup atau mati. Masukan *on-off* mengakibatkan terjadinya proses. Pada suatu pengendalian saklar listrik sehingga *system* bekerja sesuai dengan kondisi yang diinginkan, listrik menyala atau mati. Keadaan *on-off* (hidup atau mati) merupakan masukan, sedangkan mengalir dan tidak mengalir listrik merupakan keluaran. Suatu keadaan dimana listrik sudah dihidupkan

namun tidak menyala, berarti ada yang salah pada system tersebut. Proses yang dicontohkan mengilustrasikan *system* kendali yang terjadi secara manual.

Secara umum ada empat aspek yang berkaitan dengan system pengendali yaitu masukan, keluaran, system dan proses. Masukan (*input*) adalah rangsangan dari luar yang diterapkan ke sebuah *system* kendali untuk memperoleh tanggapan tertentu dari system pengaturan. Keluaran (*output*) adalah tanggapan sebenarnya yang didapatkan dari suatu system kendali.

2.1.2 Penerangan Dalam Gedung

Pada saat merancangan dalam gedung yang harus diperhatikan adalah kuat penerangan, cahaya yang diperlukan dan arah pencahayaan sumber penerangan. Kuat penerangan akan menghasilkan luminasi karena pengaruh faktor pantulan dinding maupun lantai ruangan. Pancaran cahaya perlu mendapat perhatian pada perencanaan penerangan di samping warna yang dihasilkan sumber cahaya. Sumber cahaya adalah satuan penerangan lengkap yang terdiri dari lampu beserta perlengkapan aplikasi yang lain

Sistem penerangan dari suatu sumber cahaya dipancarkan langsung ke suatu obyek penerangan atau bidang kerja. Ada 5 klasifikasi sistem pancaran cahaya dari sumber cahaya, yaitu:

1. Penerangan tak langsung

Pada penerangan tak langsung 90% hingga 100% cahaya di pancarkan ke langit-langit ruangan sehingga yang dimanfaatkan pada bidang kerja adalah cahaya pantulan.

2. Penerangan setengah tak langsung

Pada penerangan setengah tak langsung 60% hingga 90% cahaya diarahkan ke langit-langit. Distribusi cahaya pada penerangan ini mirip dengan distribusi penerangan tak langsung tetapi lebih efisien dan kuat penerangannya lebih tinggi.

3. Penerangan menyebar (difus)

Pada penerangan difus distribusi cahaya keatas dan bawah relatif merata yaitu berkisar 40% hingga 60%. Penerangan difus menghasilkan cahaya teduh dan bayangan lebih jelas dibanding yang dihasilkan dua penerangan yang dijelaskan sebelumnya.

4. Penerangan setengah langsung

Penerangan setengah langsung 60% hingga 90% cahaya diarahkan kebidang kerja selebihnya diarahkan ke langit-langit.

5. Penerangan langsung

Pada penerangan langsung 90% hingga 100% cahaya dipancarkan kebidang kerja. Pada penerangan langsung terjadi efek terowongan pada langit-langit. Yaitu tepat diatas lampu terdapat bagian yang gelap.

2.1.3 Komputer Pribadi (*Personal Computer*)

Komputer berasal dari bahasa lain *computare* yang mengandung arti menghitung. Karena luasnya bidang garapan ilmu komputer, para pakar dan peneliti sedikit berbeda dalam mendefinisikan teknologi komputer.

1. Menurut *hamacher*, komputer adalah mesin penghitung elektronik yang cepat dan dapat menerima informasi *input* digital, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya, dan menghasilkan *output* berupa informasinya
2. Menurut *blissmer*, komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas sebagai berikut :
 - a. Menerima *input*
 - b. Memproses *input* tadi sesuai dengan programnya
 - c. Menyimpan perintah-perintah dan hasil dari pengolahan menyediakan *output* dalam bentuk informasi
3. Sedangkan *fouri* berpendapat bahwa komputer adalah suatu proses data yang dapat melakukan perhitungan besar. Secara cepat, termasuk perhitungan aritmatika dan operasi logika, tanpa campur tangan dari manusia. Untuk mewujudkan konsepsi komputer sebagai pengelola data untuk menghasilkan suatu informasi, maka diperlukan sistem komputer (*computer system*) yang elemennya terdiri dari *hardware*, *software* dan *brainware*. Ketiga elemen sistem komputer tersebut harus saling berhubungan dan membentuk kesatuan. *Hardware* tidak akan berfungsi apabila tanpa *software*. Demikian juga sebaliknya. Dan keduanya tidak

bermanfaat apabila tidak ada manusia (*brainware*) yang mengoperasikan dan mengendalikannya.

2.1.4 Borland Delphi

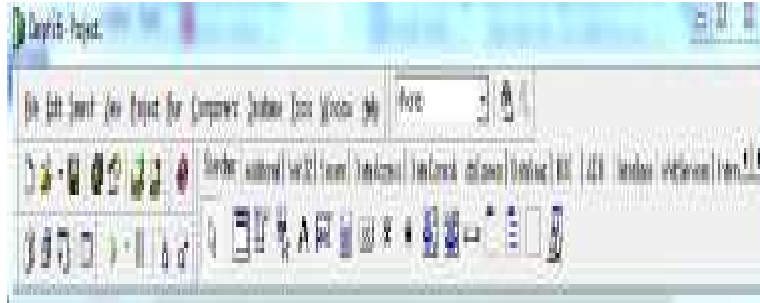
Delphi adalah Suatu bahasa pemrograman yang menggunakan visualisasi sama seperti bahasa pemrograman Visual Basic (VB). Namun Delphi menggunakan bahasa yang hampir sama dengan pascal (sering disebut object pascal). Sehingga lebih mudah untuk digunakan. Bahasa pemrograman Delphi dikembangkan oleh Code Gear sebagai divisi pengembangan perangkat lunak milik embarcadero. Divisi tersebut awalnya milik borland , sehingga bahasa ini memiliki versi Borland Delphi.

Kegunaan delphi adalah sebagai berikut:

- a. Untuk membuat aplikasi windows.
- b. Untuk merancang aplikasi program berbasis grafis.
- c. Untuk membuat program berbasis jaringan (*Client Server*).
- d. Untuk merancang program. Net (berbasis internet)

Windows utama berada pada posisi atas dari layar. *Windows* utama terdiri dari menu utama, *toolbar* dan *component palette*. Kotak judul diatas pada *windows* utama berisi nama dari proyek yang sedang dikerjakan. Kotak menu terdiri dari menu-menu tersusun kebawah.Pada bagian *Toolbar* terdapat sekumpulan *shortcut*/tombol untuk operasi-operasi yang sering digunakan

(seperti menjalankan program, menambahkan form ke sebuah proyek, menyimpan unit dan lain-lain).¹Lihat gambar 2.1.



Gambar 2.1 Window utama delphi.

2.1.4.1 Library Delphi Alpha Control

Alpha control² adalah salah satu komponen delphi yang digunakan untuk membuat tampilan aplikasi lebih menarik dengan *skins*. Sebagian besar para *programmer* memilih komponen ini karena kemudahan penggunaannya dan banyak pilihan *skins* yang bisa digunakan.

Dalam Alphacontrol sendiri juga terdapat komponen turunan dari komponen *standart* delphi yang sudah di modifikasi oleh pihak alpha skins. Berikut ini beberapa fitur yang terdapat dari komponen alpha control :²

¹http://en.wikipedia.org/wiki/Delphi_%28programming_language%29 diakses pada tanggal 23/10/15 pukul 16.20

² <http://www.alphaskins.com/apack.php> diakses pada tanggal 25/10/15 pukul 12.25

1. Mengubah *skins* (tampilan) proyek dengan mudah
2. Komponen memiliki semua fitur standart
3. *Powerfull*
4. *Skins* yang di *upgradeable* dan memiliki struktur yang sangat sederhana
5. Unsur-unsur dari setiap kontrol di hitung secara *realtime*, Sehingga kontrol apapun dapat dipindah dan di ubah ukuranya pada saat *runtime*
6. Mendukung file gambar bertipe PNG, *Talpha Image List* yang digunakan ketika *Design-time*
7. Mengontrol efek animasi
8. Mengubah *HUE* saturasi tampilan saat *run-time*.

2.1.5 PLC (*Programmable Logic Controller*)

Industri otomatis pada beberapa tahun lalu hanya menggunakan papan elektronik sebagai sistem kontrol. Penggunaan papan elektronik ini membutuhkan banyak sekali *interkoneksi* di antara relai untuk membuat agar sistem dapat bekerja. Dengan kata lain, untuk menghubungkan *relay-relay* tersebut dibutuhkan kabel yang sangat banyak. Jadi seorang ahli mesin harus membuat suatu rangkaian logika yang kemudian di implementasikan dalam

bentuk *relay*. *Relay* yang dibutuhkan dalam perancangan tersebut berjumlah ratusan dan skema yang dibuat dinamakan *ladder schematic*. *Ladder schematic* menampilkan *switch*, sensor, motor, dan *relai*. Semua peranti elektronik tersebut dihubungkan menjadi satu. Salah satu masalah yang mungkin timbul adalah jika salah satu *relay* rusak maka secara otomatis proses produksi akan berhenti dan hanya akan dapat dijalankan lagi jika *relay* tersebut telah selesai diperbaiki. Hal ini akan menyebabkan terjadinya ketidak-efisienan waktu produksi. Terkait dengan masalah ini maka munculah suatu peranti elektronika yang dapat mengatasi semua masalah tersebut, yaitu yang dinamakan PLC (*Programmable Logic Controller*). Hanya dengan mengeksekusi program yang tersimpan dalam memori, PLC dapat memonitor status dari suatu sistem berdasarkan sinyal input yang masuk pada PLC. Dalam pengontrolan suatu proses yang sangat kompleks dimungkinkan untuk menggunakan lebih dari 1 PLC.

Bentuk fisik salah satu PLC dapat dilihat pada gambar 2.2.



Gambar 2.2. PLC OMRON SYSMAC CP1E

PLC pertama kali diperkenalkan pada tahun 1960-an. PLC dibuat untuk mengurangi beban ongkos perawatan dan penggantian sistem *control* mesin yang menggunakan *relay*. Semakin meningkatnya kebutuhan dalam proses produksi menyebabkan sistem harus sering diubah-ubah. Apabila sistem yang digunakan merupakan *relay meckanic*, tentu saja hal itu akan menjadi masalah yang besar. Selain masa penggunaannya terbatas, sistem juga membutuhkan perawatan yang cermat. Jika terjadi kerusakan maka akan sangat sulit untuk menemukannya.

Oleh sebab itulah dibutuhkan pengontrol yang memudahkan baik dalam perawatan maupun penggunaannya.³

Pada tahun 70-an, teknologi PLC adalah kendali logika terprogram merupakan suatu piranti elektronik yang dirancang untuk dapat beroperasi secara digital dengan menggunakan memori sebagai media penyimpanan instruksi-instruksi internal untuk menjalankan fungsi-fungsi logika, seperti fungsi pencacah/*counter*, fungsi urutan proses, fungsi pewaktu, fungsi aritmatika, dan fungsi yang lainnya dengan cara memprogramnya. PLC sama halnya seperti mikrokontroler tapi disini PLC sudah mendapat standarisasi sebagai alat pengendali industri.

PLC dirancang untuk pengendalian proses dengan banyak */binarystate variable* (sensor) serta banyak *binary state actuator*. Basis di algoritma kontrol PLC adalah *logic IF-THEN*. Salah satu keunggulan PLC adalah mempunyai arsitektur yang *programmable*, serta *expandable* sehingga sangat adaptif untuk setiap perubahan proses serta kebutuhan sistem kontrol.

Sistem kontrol yang menggunakan PLC maupun mengontrol mesin-mesin atau proses dengan daya guna dan ketelitian yang tidak tertandingi oleh sistem kontrol konvensional yang menggunakan *relay elektro mekanis*. Keuntungan menggunakan PLC adalah sebagai berikut:

a. *Fleksibel*

³ Husanto, Thomas. ST, MT, PLC (*Programmable Logic Control*) FP Sigma, C.V andi offse, Yogyakarta, hlm 2.

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. Dapat 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.

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

2.1.5.1 Prinsip Kerja PLC

Prinsip kerja PLC dapat diuraikan sebagai berikut :PLC merupakan peralatan elektronik yang dibangun dari *mikroprosesor* untuk memonitor keadaan dari peralatan input dan kemudian dianalisa sesuai dengan kebutuhan perencanaan (*programmer*) untuk mengontrol keadaan *output*. Pada prinsipnya

PLC bekerja dengan cara menerima data-data dari peralatan input luar seperti pada gambar 2.3. 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 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.⁴

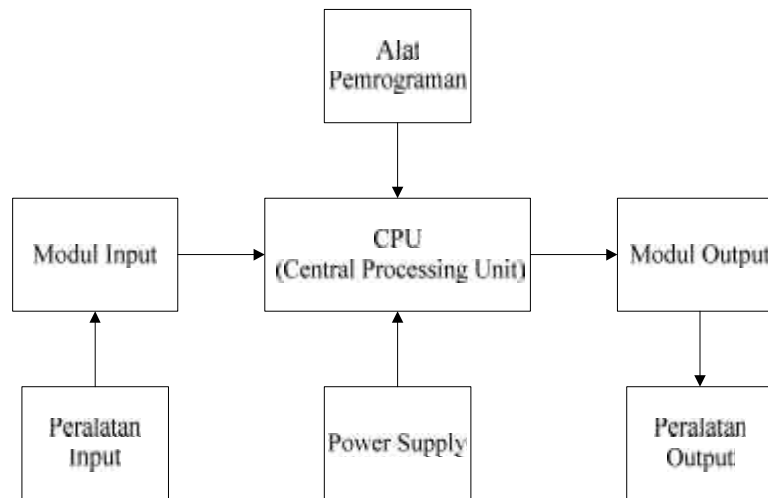


Gambar 2.3. Struktur PLC

⁴ Syufrijal. 2011. PLC. Jakarta: Universitas Negeri Jakarta.

2.1.5.2 Arsitektur PLC

Struktur PLC dapat dibagi ke dalam empat komponen utama seperti Antarmuka (*interface*) *input*, Antarmuka (*interface*) *output*, *Processing Unit* (CPU-*Central Processing Unit*), Unit memori seperti pada gambar 2.4:



Gambar 2.4. Arsitektur PLC

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 logik. 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* tersebut, PLC memanggil berdasarkan alamatnya. Untuk melaksanakan sebagai kontrol sistem, PLC ini didukung oleh perangkat lunak yang merupakan bagian penting dari PLC. Program Dan sebagai hasilnya adalah 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 *relai*.

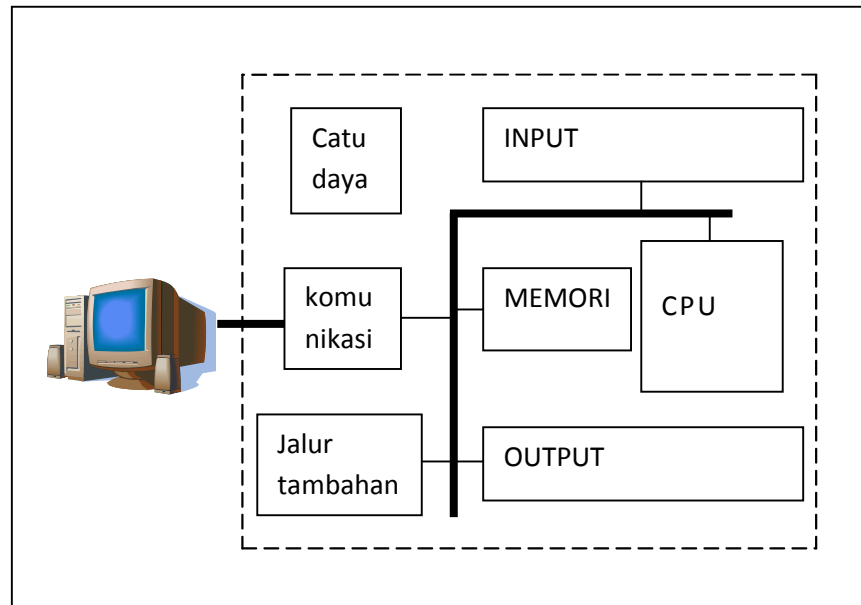
Arus informasinya dalam PLC akan mengikuti jalur yang sederhana seperti dibawah ini:

- a. CPU akan membaca “ unit memori “
- b. Memeriksa status “ Antarmuka *input* “
- c. Memperbaharui status “ CPU “
- d. Memperbaharui status “ Antarmuka *output* “

2.1.5.3. 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. 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

a. *Central Processing Unit (CPU)*

CPU berfungsi untuk mengambil instruksi dari memori, mengkodekannya 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 menangani

masalah komunikasi data dengan piranti *eksternal*, *interkonektivitas* 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 diagram *ladder* 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 memory 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

Kontroler 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 *softwarena* sendiri-sendiri. Dalam memprogram ulang PLC sangat penting diperhatikan bahwa saat sistem diperbaiki program yang benar dan sesuai harus disimpan kedalam PLC lagi. Dan juga 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, dll. 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 *input* digital saja namun untuk PLC besar mampu menerima *input analog* 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* di dalam 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*

Interfacing input 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*

Jika sistem terotomasi 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 lain sebagainya. *Outputan* 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 juga *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.

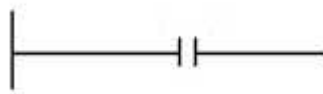
2.1.5.4. Instruksi-Instruksi Dalam Pemrograman PLC

1. Instruksi Dasar

Instruksi dasar merupakan instruksi yang digunakan untuk membuat rangkaian logika dari diagram tangga. Instruksi dasar ini ada enam, yaitu :

a. LD

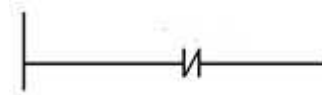
LD atau singkatan dari *Load*, merupakan instruksi untuk memulai program garis atau blok pada rangkaian logika yang dimulai dengan kontak NO (*Normally Open*) *input* seperti terlihat pada gambar 2.6.



Gambar 2.6. Instruksi LD

b. NOT

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



Gambar 2.7. Instruksi *NOT*

c. OUT

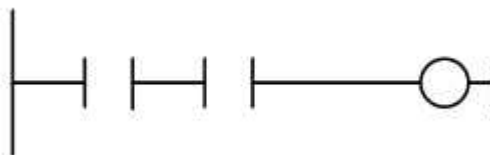
OUT merupakan instruksi untuk memasukkan program *koil output*. Kontak-kontak dari masing-masing *koil output* dapat digunakan beberapa kali sesuai yang diinginkan. Simbol instruksi *OUT* bisa dilihat pada gambar 2.8.



Gambar 2.8. Instruksi *OUT*

d. AND

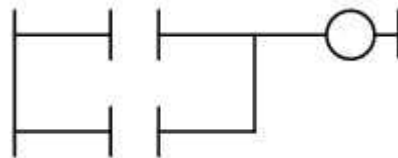
Instruksi *AND* ini digunakan untuk menghubungkan dua atau lebih kontak-kontak *input* atau *output* secara seri seperti pada gambar 2.9.



Gambar 2.9. Instruksi *AND*

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*

f. END

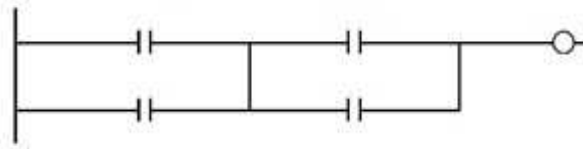
Instruksi dasar *END* untuk menyatakan rangkaian kontrol yang dibuat telah berakhir. Instruksi *END* ini harus selalu dimasukkan dalam penulisan program, karena apabila akhir rangkaian kontrol tidak dilengkapi dengan instruksi *END*, maka program tersebut tidak akan dieksekusi oleh CPU.

1. Instruksi Gabungan

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

a. *AND LD*

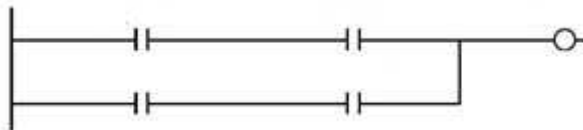
Instruksi ini merupakan gabungan dari instruksi *AND* dan *LD* yang digunakan untuk menggabungkan dua blok rangkaian dalam secara seri seperti terlihat pada gambar 2.11.



Gambar 2.11. Instruksi AND LD

b. *OR LD*

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

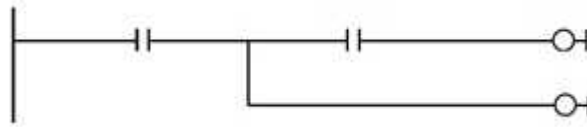


Gambar 2.12. Instruksi OR LD

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

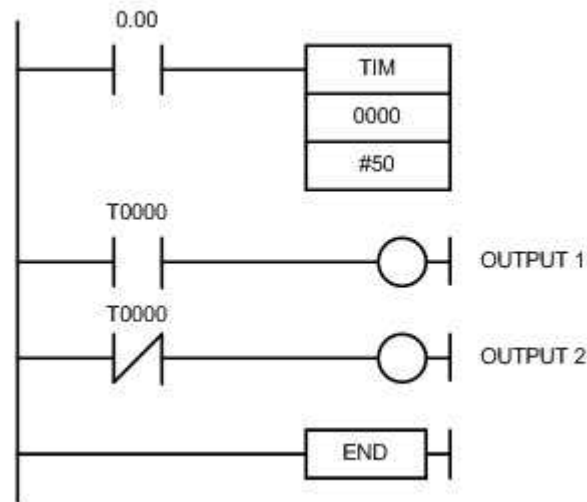
2. Instruksi Timer dan Counter

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 nomor 0 – 127. Cara kerja *timer* dan *counter* hampir mirip, perbedaannya *timer* mencacah pulsa *internal* sedangkan *counter* mencacah pulsa dari luar.

a. Timer

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 nomor dari 000 – 127 (T0 – T127). Instruksi *Timer* ada 2 macam yaitu *Timer* (TIM) dan *High Timer* (TIMH). Bedanya pada pengukuran waktu TIM 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.



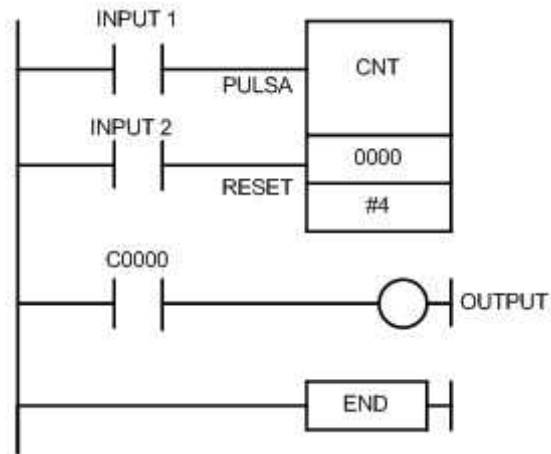
Gambar 2.14. Contoh Penggunaan *Timer*

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

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

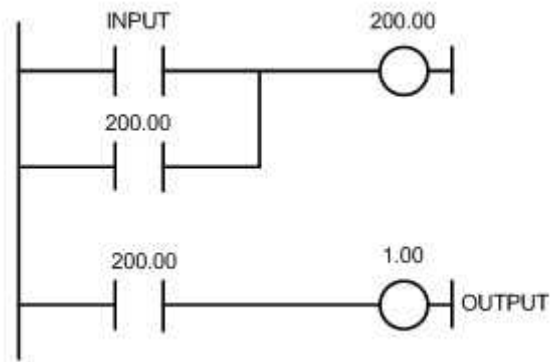
Counter akan mulai mencacah pulsa dari 0 sampai *preset value* ketika terdapat sinyal *input 1* berupa pulsa dan kondisi *input 2* sebagai *resetyaOFF*. Bila cacahan *Counter* sudah mencapai *preset value* yaitu sebanyak 4 kali maka *Counter* akan mengaktifkan *output*. Akan tetapi bila *input 2 reset ON* sebelum *Counter* mencapai *preset value* maka *Counter* akan *OFF (reset)* dan *output* akan *OFF*.

3. 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. *Internal relay* adalah *relay* semu yang merupakan bit digital yang disimpan pada *internal image register*. Dari sudut pandang pemrograman, semua PLC mempunyai satu *coil* dan mempunyai sebanyak *N/O*, *N/C* kontak sesuai yang diinginkan programmer, contoh penggunaan *internalrelay* bisa dilihat pada gambar 2.16. Semua PLC mempunyai *internal relay* akan tetapi penomoran dan jumlah maksimum yang diperbolehkan tergantung dari merek dan model PLC.

Internal relay 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 :

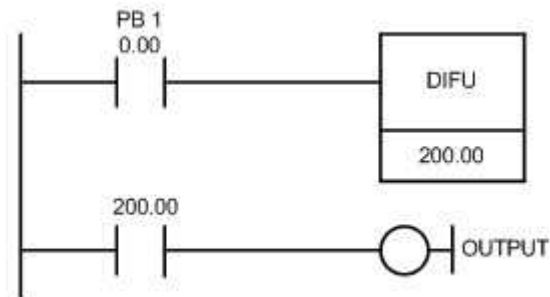
- a. 200 – 231 (*word*)
- b. 600 – 615 (*word*)
- c. 700 – 715 (*word*)
- d. 800 – 815 (*word*)
- e. 900 – 915 (*word*)



Gambar 2.16. Internal Relay

4. DIFU (*Diferential Up*)

Aplikasi kontrol ini berfungsi untuk meng *ON* kan *output* selama satu scan seperti terlihat 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.

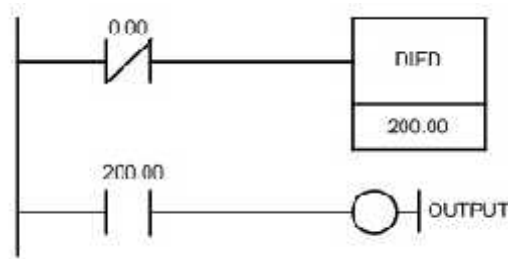


Gambar 2.17. Contoh Penggunaan DIF

6. DIFD (*Diferential Down*)

Aplikasi kontrol ini berfungsi untuk melewati 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 mati.

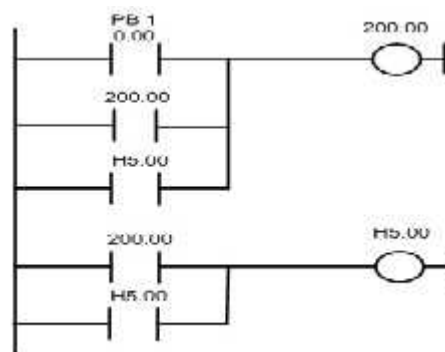


Gambar 2.18. Contoh Penggunaan DIFD

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

2.1.5.5 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. Instal 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 menginstal CX-Programmer versi terbaru. CX-Programmer dapat diinstal mulai dari OS Windows 95/98/NT 4.0 SP 6, Windows 2000/Me, hingga Windows XP dan 7.

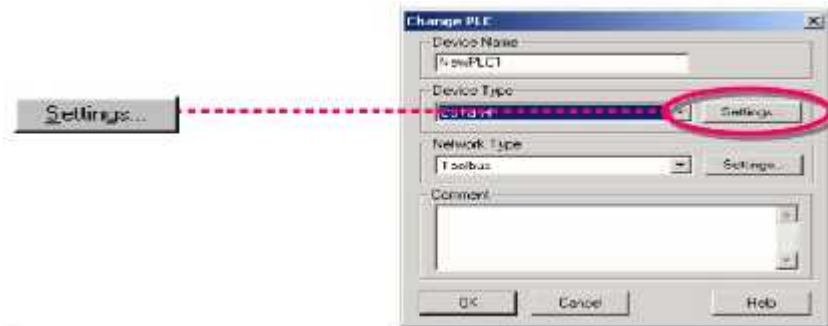
2. Membuat Project Baru

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



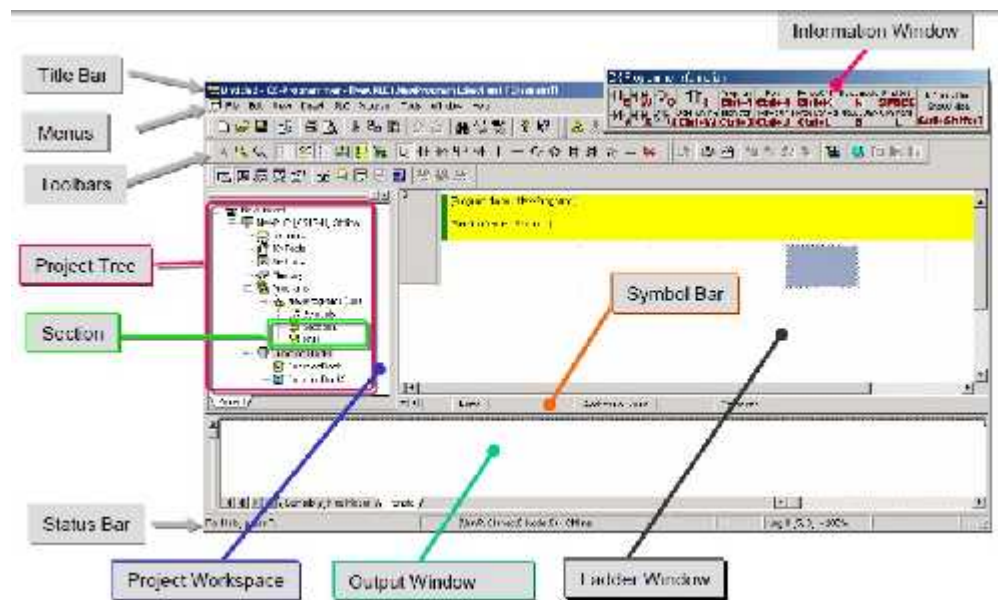
Gambar 2.20. Tombol *New* pada *Toolbar*

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

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



Gambar 2.22. Main Window Software CX-Programmer

Fungsi masing-masing menu pada gambar 2.22 dijelaskan pada tabel 2.1:

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

Nama	Fungsi
<i>Title Bar</i>	Memperlihatkan nama file yang telah di <i>save</i> pada CX-Programmer
<i>Menus</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 Project Tree</i>	Mengontrol program dan data. Dapat digunakan untuk meng- <i>copy</i> data dengan <i>Drag and Drop</i> di antara project yang berbeda atau dalam satu <i>project</i> .
<i>Ladder Window</i>	Layar untuk menulis dan mengedit diagram ladder
<i>Output Window</i>	<ul style="list-style-type: none"> a. Menunjukkan <i>error check</i> b. Menunjukkan hasil pencarian <i>contacts/coils</i> di <i>list form</i> c. Menunjukkan <i>error details</i> ketika terjadi kesalahan dalam suatu <i>file project</i>.

<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 CX-Programmer. Munculkan pilih [View] -> [Information Window].
<i>Symbol Bar</i>	Menunjukkan nama, alamat atau nilai, dan penjelasan dari simbol yang dipilih kursor.

3. Membuat Program

Untuk membuat program dengan diagram *ladder*, dapat meng-klik simbol kontak, *coil*, 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 kontak, coil, garis atau fungsi pada PLC

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 dimanupilasi ialah alamat *output*, bukan peralatan *output* secara fisik.

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

relay seluruh peralatan *input* dan *output* akan 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, *output* hanya boleh muncul 1 kali.

Seperti halnya *contact* pada *relay*, *contact* 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 *contact* pada PLC nyaris tak terbatas (hanya dibatasi oleh ketersediaan memori PLC saja).

2.1.6 *Arduino*

Arduino adalah pengendali *mikro single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*-nya memiliki prosesor Atmel AVR dan *software*-nya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para hobbyist atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler, sekaligus menawarkan berbagai macam kelebihan antara lain:

a. Murah

Papan (perangkat keras) Arduino biasanya dijual relatif murah (antara 125 ribu hingga 400 ribuan rupiah saja) dibandingkan dengan platform mikrokontroler pro lainnya. Jika ingin lebih murah lagi, tentu bisa dibuat sendiri dan itu sangat mungkin sekali karena semua sumber daya untuk

membuat sendiri Arduino tersedia lengkap di website Arduino bahkan di website-website komunitas Arduino lainnya. Tidak hanya cocok untuk *Windows*, namun juga cocok bekerja di *Linux*.

b. Sederhana dan mudah pemrogramannya

Perlu diketahui bahwa lingkungan pemrograman di Arduino mudah digunakan untuk pemula, dan cukup fleksibel bagi mereka yang sudah tingkat lanjut. Untuk guru/dosen, Arduino berbasis pada lingkungan pemrograman Processing, sehingga jika mahasiswa atau murid-murid terbiasa menggunakan Processing tentu saja akan mudah menggunakan Arduino.

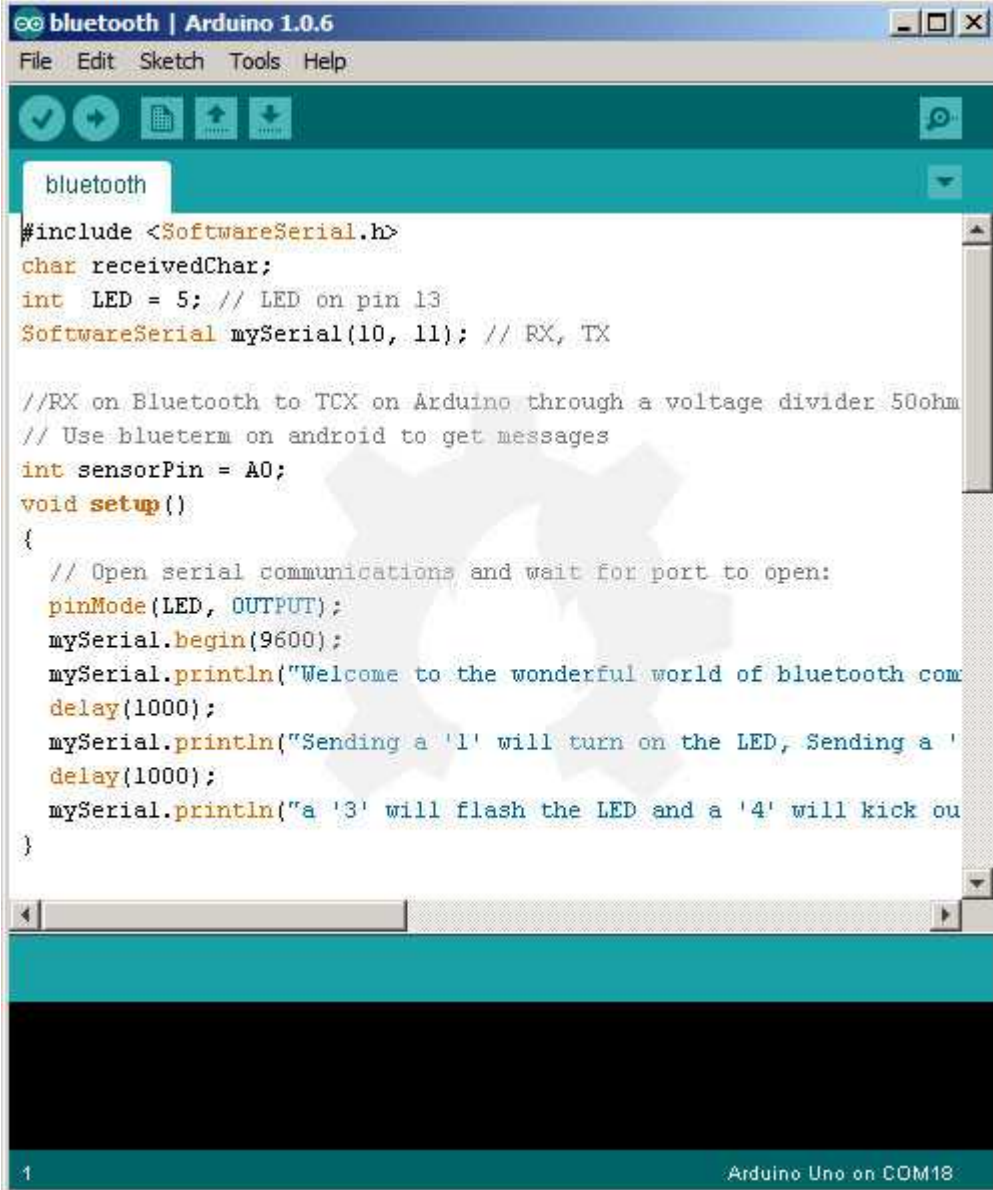
c. Perangkat lunaknya Open Source

Perangkat lunak Arduino IDE dipublikasikan sebagai Open Source, tersedia bagi para pemrogram berpengalaman untuk pengembangan lebih lanjut. Bahasanya bisa dikembangkan lebih lanjut melalui pustaka-pustaka C++ yang berbasis pada Bahasa C untuk AVR.

d. Perangkat kerasnya *Open Source*

Perangkat keras Arduino berbasis mikrokontroler ATMEGA8, ATMEGA168, ATMEGA328 dan ATMEGA1280 (yang terbaru ATMEGA2560). Dengan demikian siapa saja bisa membuatnya (dan kemudian bisa menjualnya) perangkat keras Arduino ini, apalagi bootloader tersedia langsung dari perangkat lunak Arduino IDE-nya. Bisa juga menggunakan

breadboard untuk membuat perangkat Arduino beserta perifer-al-perifer-al lain yang dibutuhkan.⁵



```

bluetooth | Arduino 1.0.6
File Edit Sketch Tools Help

bluetooth

#include <SoftwareSerial.h>
char receivedChar;
int LED = 5; // LED on pin 13
SoftwareSerial mySerial(10, 11); // RX, TX

//RX on Bluetooth to TCX on Arduino through a voltage divider 50ohm
// Use blueterm on android to get messages
int sensorPin = A0;
void setup()
{
  // Open serial communications and wait for port to open:
  pinMode(LED, OUTPUT);
  mySerial.begin(9600);
  mySerial.println("Welcome to the wonderful world of bluetooth com
  delay(1000);
  mySerial.println("Sending a '1' will turn on the LED, Sending a '
  delay(1000);
  mySerial.println("a '3' will flash the LED and a '4' will kick ou
}

1 Arduino Uno on COM18

```

Gambar 2.24. Screenshot Arduino IDE

⁵<https://ariefeeiiggeennblog.wordpress.com/pengertian-fungsi-dan-kegunaan-arduino> diakses pada tanggal 25/10/15 pukul 12.25

2.1.6.1 Kelebihan Arduino

Tidak perlu perangkat chip programmer karena didalamnya sudah ada bootloader yang akan menangani upload program dari komputer. Sudah memiliki sarana komunikasi USB, Sehingga pengguna laptop yang tidak memiliki port serial/RS323 bisa menggunakannya.

Memiliki modul siap pakai (*Shield*) yang bisa ditancapkan pada board arduino. Contohnya shield GPS, *Ethernet*, dll

.

2.1.6.2 Komponen pada Arduino

Ada banyak jenis papan Arduino yang dapat digunakan untuk tujuan yang berbeda. Beberapa papan memiliki ukuran, jumlah pin, *mikrokontroler* yang berbeda seperti pada gambar dibawah ini. Namun, sebagian besar Arduino memiliki komponen utama yang sama.

A. Daya (USB/*Barrel Jack*)

Setiap papan Arduino membutuhkan jalur untuk terhubung ke sumber listrik. Arduino Uno dapat diaktifkan melalui kable USB ang

berasal dari komputer atau *power supply* terpisah yang dihubungkan ke *Barrel Jack*. Pada gambar diatas koneksi USB diberi label (1) dan *Barrel Jack* diberi label (2).

Koneksi USB selain digunakan sebagai jalur listrik untuk mengaktifkan papan, juga digunakan untuk meng-upload kode dari komputer ke papan Arduino. Dua fungsi melalui satu jalur atau 2 in 1 lihat pada gambar 2.25

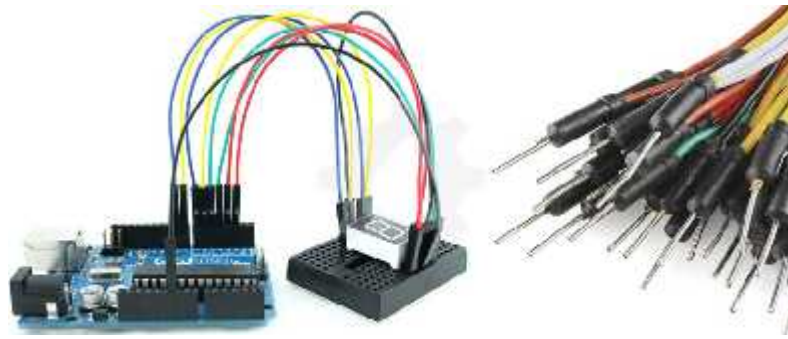


Gambar 2.25. Power Supply Barrel Jack

B. Pin (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

Pin pada Arduino adalah tempat dimana kita menghubungkan kabel untuk membuat suatu rangkaian (menghubungkan satu titik dengan titik lainnya pada breadboard dengan sejumlah kabel penghubung).

Kabel penghubung (*jumper wire*) yang digunakan dalam membuat rangkaian biasanya memiliki kepala yang terbuat dari plastik berwarna hitam yang memungkinkan kawat dari kabel penghubungnya saja yang terhubung langsung ke papan seperti gambar 2.26.



Gambar 2.26. Breadboard dan Jumper Wire

Arduino memiliki beberapa jenis pin yang masing-masing diberi label pada papan dan digunakan untuk fungsi yang berbeda-beda.

- a. **GND (3)** : **GND** merupakan singkatan dari *GROUND*. Ada beberapa pin **GND** pada Arduino, dan semuanya dapat digunakan untuk hubungan ke ground.
- b. **5V (4)** dan **3.3V (5)** : Pin 5V digunakan untuk menyediakan tegangan sebesar 5 volt. Sedangkan pin 3.3V digunakan untuk menyediakan tegangan 3,3 volt. Sebagian besar komponen sederhana yang digunakan bersama dengan Arduino berjalan pada kisaran tegangan 3,3 volt hingga 5 volt.
- c. **Analog (6)** : Pin yang berada pada area berlabel “**ANALOG IN**” (A0 sampai A5 pada Arduino Uno) digunakan sebagai pin analog. Yaitu pin yang digunakan untuk membaca sinyal-sinyal analog dari sensor-sensor analog (misal; sensor suhu) dan mengubahnya menjadi nilai digital yang dapat kita baca.

- d. **Digital (7)** : Pin digital berada disebelang pin analog (0 sampai 13 pada Arduino Uno). Pin ini dapat digunakan dalam 2 arah digital yaitu input digital (misal untuk melihat kondisi bahwa tombol sedang ditekan) dan output digital (memberikan tegangan sebuah LED).
- e. **PWM (8)** : Perhatikanlah simbol tilde (~) disamping beberapa pin digital (yaitu pada pin 3, 5, 6, 9, 10 dan 11 pada Arduino Uno). Pin ini dapat digunakan sebagai pin digital biasa, tetapi juga dapat digunakan sebagai pin PWM (*Pulse Width Modulation*). PWM biasanya digunakan sebagai pin yang mampu mensimulasikan output analog (seperti mengatur pembedaan cahaya dan warna LED saat datang dan pergi). PWM bukan analog, hanya mampu mensimulasikan analog saja. Jadi tidaklah sama antara analog dengan PWM. (Bahasan di artikel lain).
- f. **AREF (9)** : Kepanjangan dari Analog Reference atau Referensi Analog. Selama belajar, tinggalkan saja pin ini. Pin ini terkadang digunakan sebagai referensi dalam mengatur tegangan eksternal (antara 0 sampai 5 volt) untuk memberikan limit (batasan akhir) pada input pin analog.

C. Tombol Reset

Menekan tombol Reset (10) beberapa saat pada papan Arduino akan menghubungkan pin Reset ke Ground. Ini digunakan untuk me-restart kode yang telah dimuat oleh Arduino. Ini sangat bermanfaat untuk menguji ulang kode, jika kode tersebut ditulis tanpa pengulangan (repeat) atau sekali jalan. Jika anda pernah menggunakan Nintendo pasti tahu fungsi tombol Reset ini. Pada Nintendo ketika menekan tombol Reset maka hal itu digunakan untuk memperbaiki masalah, tetapi tidak demikian pada Arduino.

D. LED Indikator Daya

Tepat di bawah dan di sebelah kanan kata “UNO” pada papan Arduino Uno, terdapat sebuah LED kecil berlabel “ON” (11). LED ini harus menyala setiap kita mengubungkan papan Arduino pada sumber listrik. Jika lampu LED ini tidak menyala, kemungkinan ada papan mengalami kerusakan atau terjadi kesalahan. Periksa ulang rangkaian yang anda buat.

E. TX RX LED

TX adalah singkatan dari ‘*transmit*’ (kiriman), dan RX adalah singkatan dari ‘*receive*’ (terima). LED indikator TX dan RX (12) ini akan berkedip redup atau terang dengan jeda tak tentu untuk memberitahukan bahwa telah terjadi komunikasi serial. Kedipan LED ini sebagai indikasi visual yang merupakan pertanda baik bahwa telah terjadi pengiriman dan

penerimaan data pada papan Arduino (misal ketika kita meng-upload kode baru ke dalam papan).

F. IC Utama

Sesuatu yang berwarna hitam dengan semua kakinya terbuat dari logam inilah yang disebut IC atau *Integrated Circuit* (13). Ini adalah otak dari papan Arduino. IC utama Arduino berbeda-beda sesuai dengan jenis papan, tetapi biasanya memiliki ciri khas bertuliskan ATmega yang merupakan IC buatan perusahaan ATMEL. Ini merupakan hal penting, karena kita perlu untuk memahami jenis IC (sesuai juga dengan jenis papan Arduino) sebelum memuat atau meng-upload sekumpulan kode program yang telah kita buat dari perangkat lunak Arduino IDE ke dalam papan Arduino. Informasi tentang tipe IC dapat kita temukan pada permukaan IC tersebut. Jika kita ingin tahu lebih jauh tentang perbedaan jenis IC yang digunakan pada keluarga Arduino maka sebaiknya selalu unduh datasheet dari masing-masing tipe IC.

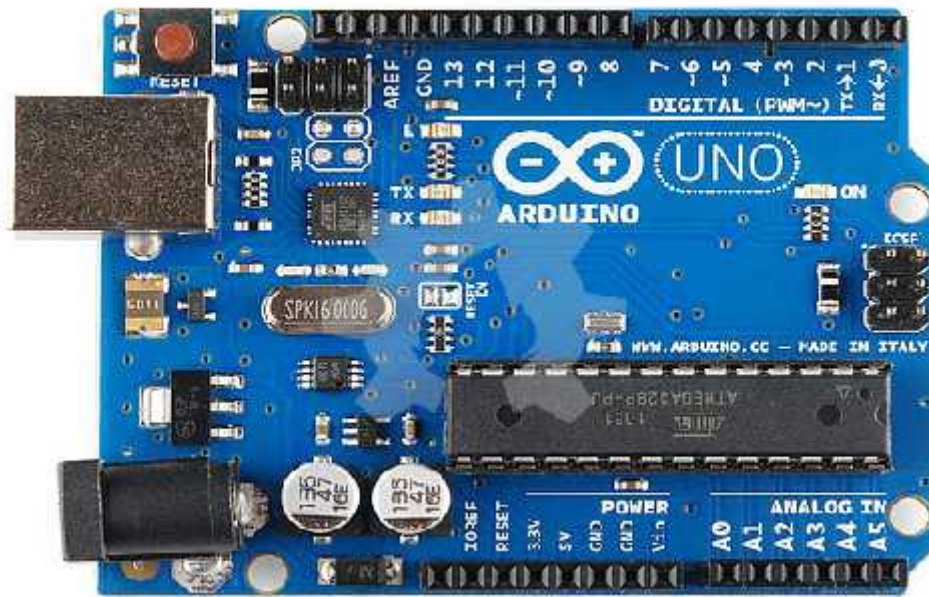
G. Regulator Tegangan

Voltage Regulator (14) bukanlah sesuatu yang mampu berinteraksi dengan papan Arduino. Tetapi kita harus mengetahui fungsi dari voltage regulator. *Voltage Regulator* atau *Regulator* Tegangan akan mengalirkan sejumlah tegangan teregulasi ke dalam papan Arduino. *Voltage Regulator* berperan sebagai penjaga pintu gerbang, dia akan

membalikan atau membuang tegangan berlebihan yang bisa membahayakan rangkaian. Tapi tentu saja ada batasnya, pastikan bahwa tegangan yang masuk ke dalam papan tidak lebih besar dari 20 volt.

2.1.6.3. Arduino Uno

Arduino Uno adalah pilihan yang tepat bagi mereka yang baru pertama kali ingin mempelajari Arduino. Karena Uno merupakan paket lengkap untuk memulai belajar Arduino. Memiliki 14 pin *input/output* digital (dimana 6 pin dapat digunakan sebagai output PWM), 6 pin *input analog*, koneksi USB, jack daya, tombol reset, dan masih banyak lagi. Ini semua sudah cukup untuk keperluan belajar mikrokontroler. Hanya dengan menghubungkan papan Arduino ke komputer melalui kabel USB, atau menggunakan adaptor AC-DC, atau menggunakan baterai untuk mengaktifkan papan Arduino. Seperti gambar 2.27



Gambar 2.27. Papan Arduino Uno

2.1.7 *Wireless Network*



Gambar 2.28. Wireless dan fungsinya

Wireless network atau **jaringan tanpa kabel** adalah salah satu jenis jaringan berdasarkan media komunikasinya, yang memungkinkan perangkat-perangkat didalamnya seperti komputer, hp, dll. Bisa saling berkomunikasi secara *wireless* atau tanpa kabel. *Wireless network* umumnya di implementasikan menggunakan komunikasi radio. Implementasi ini berada pada level lapisan fisik (*physical layer*) dari OSI model.

2.1.7.1 Tipe-tipe Wireless Network

a. *Wireless* PAN (WPAN)

Wireless Personal Area Network (WPAN) adalah jaringan *wireless* dengan jangkauan area yang kecil. Contohnya Bluetooth, *Infrared*, dan ZigBee.

b. *Wireless* LAN (WLAN) / Wifi

Wireless Local Area Network (WLAN) atau biasa disebut Wifi memiliki jangkauan yang jauh lebih luas dibanding WPAN. Saat ini WLAN mengalami banyak peningkatan dari segi kecepatan dan luas cakupannya. Awalnya WLAN ditujukan untuk penggunaan perangkat jaringan lokal, namun saat ini lebih banyak digunakan untuk mengakses internet.

c. *Wireless* MAN (WMAN)

Wireless Metropolitan Area Network (WMAN) adalah jaringan *wireless network* yang menghubungkan beberapa jaringan WLAN. Contoh teknologi WMAN adalah WiMAX.

d. *Wireless* WAN (WWAN)

Wireless Wide Area Network adalah jaringan *wireless* yang umumnya menjangkau area luas misalnya menghubungkan kantor pusat dan cabang antar provinsi.

e. *Cellular Network*

Cellular Network atau *Mobile Network* adalah jaringan radio terdistribusi yang melayani media komunikasi perangkat mobile seperti handphone, pager, dll. Contoh sistem dari *Cellular Network* ini adalah GSM, PCS, dan D-AMPS.

2.1.7.2 *Wireless* ESP8266

ESP8266 *WiFi Module* adalah mandiri SOC terintegrasi dengan TCP / IP *stack* protokol yang dapat memberikan akses ke mikrokontroler jaringan WiFi Anda. ESP8266 mampu baik *hosting* aplikasi atau pembongkaran semua fungsi jaringan Wi-Fi dari prosesor aplikasi lain. Setiap modul ESP8266 datang pra-diprogram dengan perintah AT *set firmware*, makna, Anda hanya dapat menghubungkan hal ini ke perangkat Arduino Anda dan mendapatkan sekitar sebanyak WiFi-kemampuan sebagai WiFi Perisai penawaran (dan itu hanya keluar dari kotak)! Modul ESP8266 adalah sangat biaya efektif papan dengan komunitas besar dan pernah berkembang,.

Modul ini memiliki pengolahan dan penyimpanan kemampuan yang kuat cukup *on-board* yang memungkinkan untuk diintegrasikan dengan sensor dan perangkat khusus aplikasi lain melalui GPIOs dengan pengembangan minimal muka dan minimal memuat selama *runtime*. Tingkat tinggi integrasi on-chip memungkinkan untuk sirkuit *eksternal* minimal, termasuk modul *front-end*, dirancang untuk menempati daerah PCB minimal. ESP8266 mendukung RSDP untuk aplikasi *VoIP* dan Bluetooth interface *co-eksistensi*, berisi RF diri dikalibrasi memungkinkan untuk bekerja di bawah semua kondisi operasi, dan tidak memerlukan bagian RF *eksternal*.

Hampir tak terbatas dari informasi yang tersedia untuk ESP8266, semua yang telah disediakan oleh dukungan masyarakat yang luar biasa. Dalam Dokumen bagian bawah Anda akan menemukan banyak sumber daya untuk membantu Anda dalam menggunakan ESP8266, bahkan petunjuk tentang cara untuk mengubah modul ini menjadi sebuah IOT (*Internet of Things*)

2.1.8. Relai

Relai adalah suatu rangkaian *switching* magnetik yang bekerja bila mendapat catu dari rangkaian *trigger*. Relai memiliki tegangan dan arus nominal yang harus dipenuhi *output* rangkaian *driver*-nya/pengemudinya. Arus yang digunakan pada rangkaian adalah arus DC.

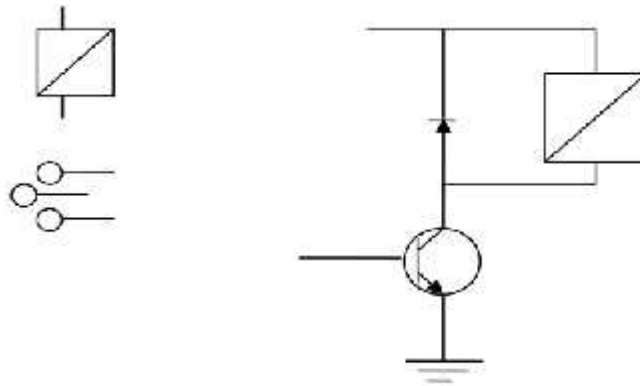
Konstruksi dalam suatu relai terdiri dari lilitan kawat (*coil*) yang dililitkan pada inti besi lunak. Jika lilitan kawat mendapatkan arus, inti besi lunak menghasilkan medan magnet dan menarik *switch* kontak. *Switch* kontak mengalami gaya tarik magnet sehingga berpindah posisi ke kutub lain atau terlepas dari kutub asalnya. Keadaan ini akan bertahan selama arus mengalir pada kumparan relai. Dan relai akan kembali keposisi semula itu *normally-off*, bila tidak ada lagi arus yang mengalir padanya. Posisi normal relai tergantung pada jenis relai yang digunakan. Dan pemakaian jenis relai tergantung pada keadaan yang diinginkan dalam suatu rangkaian/sistem.

Menurut kerjanya relai dapat dibedakan menjadi:

1. *Normaly Open* (NO); saklar akan tertutup bila dialiri arus.
2. *Normaly Close* (NC); saklar akan terbuka bila dialiria rus.
3. *Change Over*(CO); relai ini mempunyai saklar tunggal yang normalnya tertutup

Yang mana bila kumparan 1 dialiri arus maka saklar akan terhubung keterminal A, sebaliknya bila kumparan 2 dialiri arus maka saklar akan terhubung keterminal B. Analogi rangkaian relai yang digunakan adalah saat basis transistor ini dialiri arus maka transistor dalam keadaan tertutup yang dapat menghubungkan arus dari kolektor ke emiter yang mengakibatkan relai terhubung. Sedangkan fungsi dioda di sini adalah untuk melindungi transistor

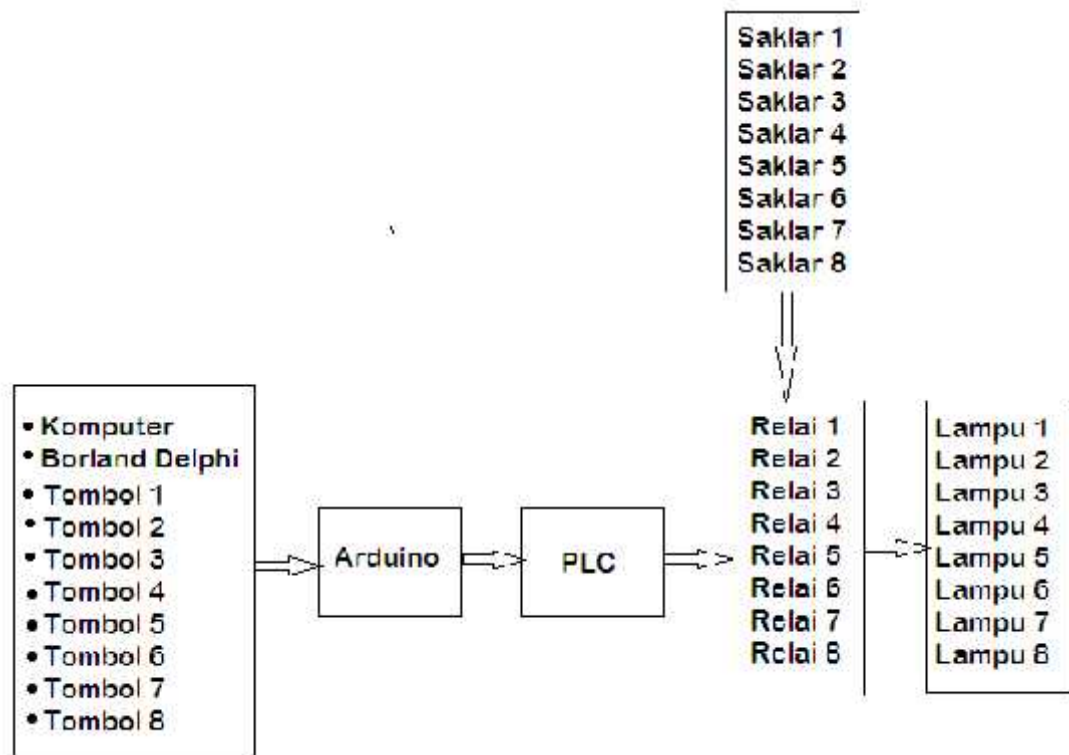
dari tegangan induksi yang bisa mencapai 100 sampai 150 volt di mana tegangan ini dapat merusak transistor lihat di gambar 2.29.



Gambar 2.29 Simbol Relai dan Relai dalam Rangkaian

2.2 Kerangka Berpikir

Adapun hubungan komunikasi antara perangkat sistem PLC dengan peralatan *output*-nya, yakni lampu-lampu ruangan digambarkan pada diagram blok Pemanfaatan PLC pada alat pengendali lampu ruangan secara terprogram, dimaksudkan untuk memberi kemudahan pada pengguna (*user*) dalam mengoperasikan/mengatur nyala (*on*) dan mati (*off*) lampu di beberapa ruangan berada secara otomatis berikut 2.30.



Gambar 2.30 Diagram Blok Alat

Gambar 2.30 menerangkan alur kerja antara *hardware* dan juga antar blok. Blok *interface* disini merupakan sebagai blok *input* yang menggunakan fungsi dari *serial port* pada *borland delphi* dan juga sebagai *monitoring* sistem. Sinyal yang dikirimkan melalui *serial port* dari *interface* akan diterima dan dibaca oleh PLC sebagai blok proses dan *controller*.

Blok proses merupakan blok yang berfungsi untuk mengolah masukan dari *serial input*. PLC merupakan inti dari blok diagram proses. PLC bertugas merespon dan mengolah masukan yang diterima dari *serial port* yang terhubung dengan *interface*.

Blok *driver* relai merupakan blok *output*. Data yang telah diolah oleh PLC akan diteruskan ke *driver* relai sebagai *output*. Relay disini berfungsi sebagai *driver* untuk sumber tegangan listrik PLN 220 V.