

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manusia umumnya mengenal sumber mata air berasal dari 3 sumber, yaitu sumber dari air permukaan, air tanah, dan air hujan. Air hujan tercipta karena penguapan air laut, air tanah tercipta karena air permukaan yang meresap ke dalam tanah, sedangkan air permukaan tercipta karena tergenangnya air hujan yang jatuh ke daratan.

Air permukaan dan air tanah merupakan sumber air alternatif bagi manusia, karena mudah di dapat dan juga sangatlah ekonomis. Air laut adalah salah satu contoh dari air permukaan yang sering dipergunakan. Air laut sangatlah berpengaruh terhadap perkembangan teknologi, pengaruh teknologi yang di maksud adalah pada sistem kerja pompa air. Pompa air harus bekerja dengan menyesuaikan ketinggian air laut yang terus berubah – ubah. Ketinggian air yang terus berubah yang mengakibatkan kerusakan dan *overheat* pada pompa air. Pengaruh tersebut disebabkan karena penyedotan air laut secara besar-besaran dan terus – menerus yang mengakibatkan pompa air tidak dapat bekerja secara maksimal.

Solusi yang dapat diterapkan untuk mengatasi masalah tersebut adalah melakukan pengontrolan sekuensial pada pompa air. Dengan mengatur pompa bekerja secara berurutan dan sesuai dengan spesifikasi pompa itu sendiri. Pengontrolan sekuensial pada pompa air dilakukan dengan menggunakan metode *Grafcet*. Metode *grafcet* adalah sebuah bahasa pemrograman berbasis grafik yang merepresentasikan program kontrol berupa langkah-langkah dari mesin atau proses. *Grafcet* ini dijadikan sebagai dasar bahasa pemrograman standar dan mudah dimengerti (Rusli, M.,2012). Diharapkan dengan menggunakan kontrol sekuensial dan metode *grafcet*, sistem pompa air akan menjadi lebih baik dan maksimal.

Solusi selanjutnya yaitu digunakannya *Programmable Logic Controller* (PLC), karena PLC adalah sebuah rangkaian elektronik yang dapat mengerjakan berbagai fungsi-fungsi kontrol pada level - level yang kompleks. PLC juga dapat diprogram, dikontrol, dan dioperasikan oleh operator yang tidak berpengalaman dalam mengoperasikan komputer. PLC akan mengoperasikan semua hasil sistem yang menghasilkan *output*, haruskah berlogika *on* atau *off* dan dapat juga dioperasikan suatu sistem dengan *output* yang bervariasi, oleh karena itu PLC sangatlah dibutuhkan untuk mengontrol sebuah pompa air agar bekerja lebih baik dan maksimal.

Pada skripsi ini akan dibuat suatu desain alat *smart pump plant* dengan kombinasi 8 pompa dan 3 kondisi *input*. *Smart pump plant* adalah sebuah *plant* berbentuk alat yang mengontrol pompa dengan kondisi *input* yang diharapkan, yaitu pompa air akan mati jika sensor suhu mendeteksi $>50^{\circ}\text{C}$, kemudian pompa air akan aktif jika sensor *limit switch* terkena oleh air dengan ketinggian 3 cm, 6 cm, dan 9 cm, yang terakhir adalah pompa akan mati setelah pompa bekerja selama 60 detik. Ketiga kondisi *input* inilah yang menjadikan sebuah alat *smart pump plant*, yang dimana alat ini akan dikontrol dengan pengontrolan sekuensial. Pengontrolan sekuensial digunakan agar menghasilkan *output* yang diharapkan, yaitu pompa air dapat bekerja secara baik dan maksimal.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang telah diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada point berikut:

1. Bagaimana perancangan program kontrol sekuensial PLC dengan metode *grafcet* pada miniature/prototype *Smart Pump Plant* ?
2. Bagaimana pengaplikasian program PLC pada miniatur/prototype *Smart Pump Plant* ?

1.3 Batasan Masalah

1. *Smart Pump Plant* yang digunakan adalah miniatur/prototipe dengan desain sendiri.
2. Kontrol sekuensial lebih ditekankan dalam skripsi.
3. Kerja motor pompa air konstan.
4. Hanya membahas pengontrolan pada motor pompa air.
5. Sensor yang digunakan untuk mendeteksi temperatur pompa air adalah sensor LM35
6. Sensor yang digunakan untuk mendeteksi level air adalah sensor *limit switch*.
7. Motor pompa air yang digunakan adalah motor pompa air DC 24 V dengan type TIY-152.
8. Pipa yang di gunakan untuk menyuplai air ke dalam tandon air penyimpanan berupa selang dengan ukuran diameter selang 2 inci.
9. Prinsip kerja sistem kontrol PLC secara umum dan mendasar.
10. Bahasa pemrograman PLC yang digunakan adalah CX-Program.

1.4 Tujuan

Mengontrol miniatur/prototype *Smart Pump Plant* dengan menggunakan sistem kontrol sekuensial dan metode *grafcet* berbasis PLC.

1.5 Sistematika Pembahasan

Adapun sistematika penulisan skripsi ini adalah:

BAB I PENDAHULUAN

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Menjelaskan dasar teori yang mendukung dalam pembuatan alat dan perancangan algoritma yang terdiri atas PLC, metode *grafcet*, motor pompa DC 24 V, sensor suhu LM35, sistem kontrol otomatis, sistem kontrol sekuensial, program CX-One, sensor *limit switch*, *power supply*, pompa sentrifugal, dan *relay*.

BAB III METODE PENELITIAN

Bab ini berisi tentang prinsip kerja alat, diagram state, tabel *input* dan *output*, perancangan pembuatan alat, metode *grafcet*, dan ladder diagram. .

BAB IV HASIL DAN PEMBAHASAN

Membahas hasil pengujian sistem, simulasi sistem dan analisis data secara keseluruhan terhadap alat yang telah direalisasikan.

BAB V KESIMPULAN DAN SARAN

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perancangan algoritma dan pengujian alat serta saran yang diperlukan untuk dilakukan pengembangan selanjutnya.



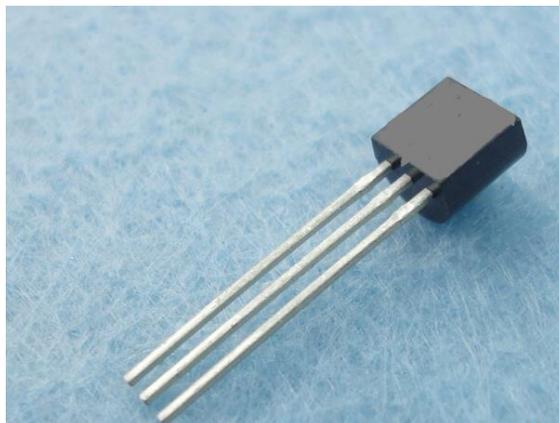
BAB II

TINJAUAN PUSTAKA

2.1 Sensor Temperatur LM35

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 merupakan komponen elektronika berbentuk *integrated circuit* (IC) dengan 3 pin yang diproduksi oleh *National Semiconductor*. Sensor suhu LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, sensor suhu LM35 juga mempunyai *output* impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kontrol khusus.

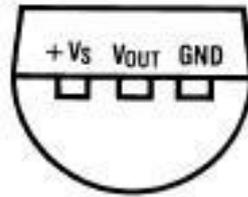
Tegangan sensor suhu LM35 ini dapat mencapai 30 volt akan tetapi yang diberikan ke sensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60 μA . LM35 mempunyai kemampuan menghasilkan panas (*self-heating*) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0,5 $^{\circ}\text{C}$ pada suhu 25 $^{\circ}\text{C}$ (Tri, A., 2011). Sensor LM35 dapat dilihat dalam Gambar 2.1.



Gambar 2.1 Sensor temperatur LM35

LM35 mempunyai 3 pin dengan fungsi masing-masing diantaranya yaitu pin 1 berfungsi sebagai sumber tegangan kerja dari LM35, pin 2 atau tengah digunakan sebagai tegangan keluaran yaitu dari 0 volt sampai dengan 1,5 volt, sedangkan pin 3 adalah *ground*. Sensor dapat dilihat dalam Gambar 2.2.

TO-92
Plastic Package



BOTTOM VIEW

Gambar 2.2 Bagian Sensor LM35

(Sumber : Tri, A., 2011)

Tegangan operasi sensor LM35 yang dapat digunakan antara 4 volt sampai 30 volt. Keluaran sensor ini akan naik sebesar 10 mV setiap derajat *celcius* sehingga diperoleh persamaan sebagai berikut :

$$V_{LM35} = \frac{10mV}{^{\circ}C}$$

Secara prinsip sensor akan melakukan perubahan suhu setiap suhu 1 °C dan akan menunjukkan tegangan sebesar 10 mV. Pada penempatannya LM35 dapat ditempelkan dengan perekat pada permukaan akan tetapi suhunya akan sedikit berkurang sekitar 0,01 °C, karena terserap pada suhu permukaan tersebut. Dengan cara seperti ini diharapkan selisih antara suhu udara dan suhu permukaan dapat dideteksi oleh sensor LM35 sama dengan suhu disekitarnya, jika suhu udara disekitarnya jauh lebih tinggi atau jauh lebih rendah dari suhu permukaan, maka LM35 berada pada suhu permukaan dan suhu udara disekitarnya. Pada seri LM35 $V_{out} = 10 \text{ mV}/^{\circ}C$, dalam arti tiap perubahan 1 °C akan menghasilkan perubahan tegangan output sebesar 10 mV.

2.2 Programmable Logic Control (PLC)

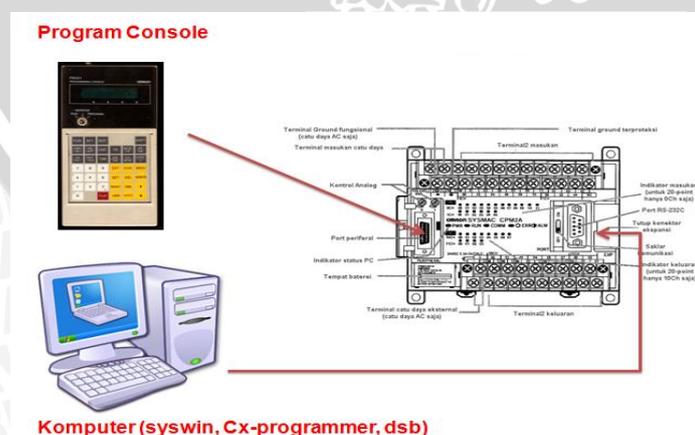
Programmable Logic Controller (PLC) adalah sebuah rangkaian elektronik yang dapat mengerjakan berbagai fungsi-fungsi kontrol pada level - level yang kompleks. PLC dapat diprogram, dikontrol, dan dioperasikan oleh operator yang tidak berpengalaman dalam mengoperasikan komputer. PLC umumnya digambarkan dengan garis dan peralatan pada suatu *ladder diagram*. *Ladder diagram* tersebut pada komputer menggambarkan hubungan yang diperlukan untuk suatu proses. PLC akan mengoperasikan semua sistem yang

mempunyai *output* haruskah berlogika *on* atau *off* dan dapat juga dioperasikan suatu sistem dengan *output* yang bervariasi.

PLC pada awalnya sebagai alat elektronik untuk mengganti panel *relay*. Pada saat itu PLC hanya bekerja untuk kondisi *on - off* untuk pengendalian motor, *solenoid*, dan *actuator*. Alat ini mampu mengambil keputusan yang lebih baik dibandingkan *relay*. Pertama kali PLC banyak digunakan pada bagian otomotif. Sebelum adanya PLC sudah banyak peralatan kontrol sekuensial, ketika *relay* muncul, panel kontrol dengan *relay* menjadi kontrol sekuensial yang utama. Ketika transistor muncul, *relay* yang diterapkan yaitu untuk mengontrol dengan kecepatan tinggi. Pada tahun 1978, penemuan *chip mikroprocessor* menaikkan kemampuan komputer untuk segala jenis sistem otomatisasi dengan harga yang terjangkau. Program PLC makin mudah untuk dimengerti oleh banyak orang (Effendi, A., 2013).

Pada awal tahun 1980 PLC makin banyak digunakan di beberapa perusahaan elektronik dan komputer untuk membuat PLC dalam jumlah yang banyak. Sekarang sistem kontrol sudah meluas dan sistem kontrol dikombinasikan dengan *feedback control*, pemrosesan data, dan sistem monitor terpusat. Saat ini PLC sudah menjadi alat yang cerdas, yang merupakan kebutuhan utama di industri modern.

PLC modern juga sebagai alat yang dapat mengeksekusi data dan menyimpannya. PLC sebenarnya adalah suatu sistem elektronika digital yang dirancang agar dapat mengendalikan mesin dengan proses mengimplementasikan fungsi nalar kontrol sekuensial, operasi pewaktuan (*timing*), pencacahan (*counting*), dan aritmatika. PLC adalah komputer digital sehingga, mempunyai bagian – bagian pendukung seperti *processor*, unit memori, unit kontrol, dan unit I/O (Effendi, A., 2013) seperti dalam Gambar 2.3.



Gambar 2.3 Sebuah pengontrol logika terprogram
(Sumber : Effendi, A., 2013).

Kehandalan PLC yang menjadikannya lebih baik dibandingkan *relay* (Effendi, A., 2013) yaitu :

a. *Flexibility*

Pada awalnya, setiap mesin produksi yang dikendalikan secara elektronik memerlukan pengontrolan, misalnya 12 mesin memerlukan 12 kontroler. Sekarang dengan menggunakan satu model dari PLC sudah dapat mengontrol salah satu dari 12 mesin tersebut, dengan tiap mesin dikontrol dengan masing-masing program.

b. Perubahan implementasi dan koreksi eror

Dengan menggunakan tipe *relay* yang terhubung pada panel, perubahan program akan memerlukan waktu untuk menghubungkan kembali menuju panel dan peralatan. Sedangkan dengan menggunakan PLC untuk melakukan perubahan program, tidak memerlukan waktu yang lama yaitu dengan cara merubahnya pada sebuah *software*. Jika kesalahan program terjadi, maka kesalahan dapat langsung terdeteksi keberadaannya dengan memonitor secara langsung. Merubahnya juga sangatlah mudah yaitu hanya dengan mengubah *ladder diagram*-nya.

c. Harga yang rendah

PLC lebih sederhana karena bentuk, ukuran, dan peralatan lain yang mendukungnya, sehingga harganya dapat dijangkau. Saat ini di dalam sebuah PLC sudah terdapat *timer*, *counter*, dan *input analog* di dalam satu kemasan CPU. PLC mudah di dapat dan kini sudah banyak beredar di pasaran dengan bermacam-macam tipenya.

d. Jumlah kontak yang banyak

PLC memiliki jumlah kontak yang banyak untuk tiap koil yang tersedia. Contohnya yaitu panel penghubung *relay* yang mempunyai 5 kontak dan semuanya digunakan, sementara pada perubahan desain tersebut diperlukan 4 kontak lagi yang berarti diperlukan penambahan satu buah *relay* lagi. Ini berarti diperlukan waktu untuk melakukan instalasinya, dengan menggunakan PLC hanya diperlukan pengetikan untuk membuat 4 buah kontak. Ratusan kontak dapat digunakan dari satu buah *relay*, jika memori pada komputer masih memungkinkan.

e. Hasil tampilan

Rangkaian program PLC dapat dicoba terlebih dahulu, diuji, diteliti dan dimodifikasi pada kantor atau laboratorium, sehingga efisiensi waktu dapat dicapai. Untuk menguji program PLC tidak harus diinstalasikan dahulu menggunakan alat yang hendak dijalankan, tetapi dapat dilihat langsung pada CPU PLC atau dilihat pada *software* pendukungnya.

f. Observasi visual

Operasi dari rangkaian PLC dapat dilihat selama dioperasikan secara langsung melalui layar. Jika ada kesalahan operasi maupun kesalahan yang lain dapat langsung diketahui, maka jalur logika akan menyala pada layar sehingga perbaikan dapat lebih cepat dilakukan.

g. Kecepatan operasi

Kecepatan operasi dari PLC melebihi kecepatan operasi *relay* pada saat bekerja yaitu dalam beberapa detik, sehingga dapat menentukan kecepatan *output* dari alat yang digunakan.

h. *Bolean metode or ladder*

Program PLC dapat dilakukan dengan *ladder diagram* oleh para teknisi atau juga menggunakan *boolean system* atau digital bagi para pemrogram, karena PLC lebih mudah dan dapat disimulasikan pada *software* pendukungnya.

i. *Reliability*

Peralatan *solid state* umumnya lebih tahan dibandingkan dengan *relay* atau *timer* mekanik. PLC mampu bekerja pada kondisi lingkungan yang berat, misalnya guncangan, debu, suhu yang tinggi, dan sebagainya.

Fungsi dan kegunaan dari PLC dapat dikatakan hampir tidak terbatas, tetapi dalam prakteknya dapat dibagi secara umum dan khusus (Haase, K, 2010). Fungsi dari PLC adalah sebagai berikut :

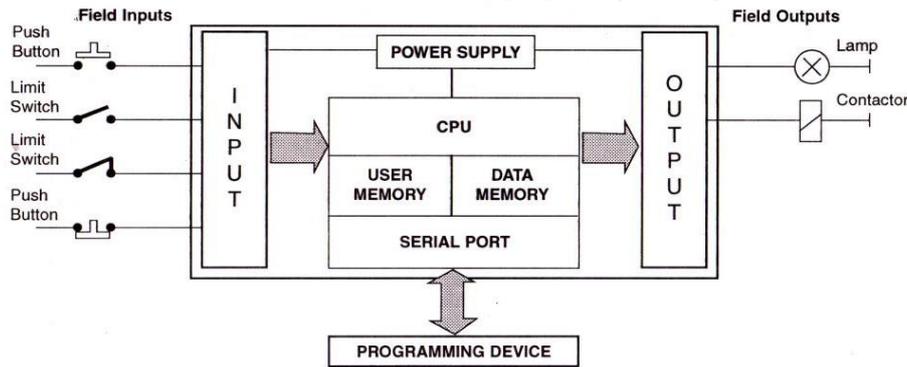
1. Kontrol Sekuensial

PLC memproses *input* sinyal biner menjadi *output* yang digunakan untuk keperluan pemrosesan teknik secara berurutan (sekuensial), disini PLC menjaga agar semua *step* dalam proses sekuensial yaitu berlangsung dalam urutan yang tepat.

2. *Monitoring Plant*

PLC secara terus menerus memonitor suatu sistem (misalnya temperatur, tekanan, tingkat ketinggian dan mengambil tindakan yang diperlukan. *Monitoring* juga sehubungan dengan proses yang dikontrol (misalnya nilai sudah melebihi batas) atau menampilkan pesan tersebut ke operator.

Sistem PLC memiliki tiga komponen utama yaitu unit *processor*, bagian *input / output*, dan perangkat pemrograman. Fungsi kerja dari ketiga komponen tersebut dapat dilihat dalam Gambar 2.4.



Gambar 2.4 Bagian komponen utama PLC Omron CP1L
(Sumber :Haase, K., 2010)

a. Unit *Processor*

Processor adalah bagian pemrosesan dari sistem PLC yang akan membuat keputusan logika. Keputusan yang telah dibuat berdasarkan pada program yang telah disimpan pada memori. *Processor* adalah bagian dari *Central Processing Unit* (CPU) dari PLC yang akan menerima, menganalisa, memproses dan memberikan informasi ke modul *output*. Di dalam CPU PLC dapat dibayangkan seperti kumpulan dari ribuan *relay*. Hal tersebut bukan berarti di dalamnya terdapat banyak *relay* dalam ukuran yang sangat kecil, tetapi berisi rangkaian elektronika digital yang dapat difungsikan sebagai kontak NO dan NC pada *relay* (Prayogo, R., 2012).

Memori berfungsi sebagai tempat dimana informasi tersebut disimpan. Ada bermacam-macam jenis memori dalam bentuk *Integrated Circuits* (IC). Salah satu jenis memori yang digunakan dalam CPU PLC adalah *Random Access Memory* (RAM). Satu kerugian dari jenis memori tersebut adalah diperlukannya catu daya untuk menjaga agar memori tetap bekerja. Pada aplikasi PLC diperlukan catu daya cadangan yang digunakan untuk menjaga agar isi dari memori tidak hilang apabila tiba-tiba catu daya hilang.

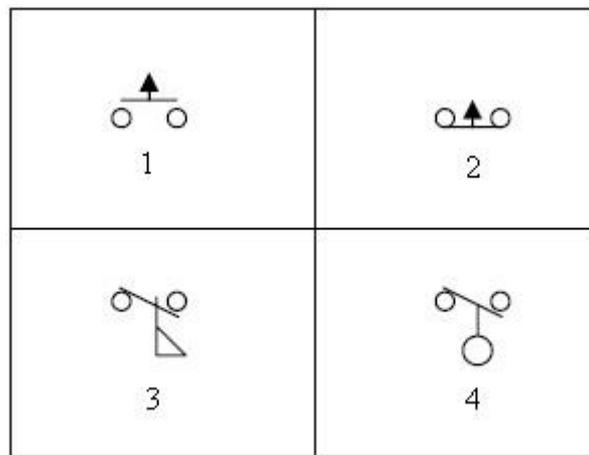
Read Only Memory (ROM) adalah jenis memori yang semi permanen dan tidak dapat diubah dengan pengubah program. Memori tersebut hanya digunakan untuk membaca saja dan jenis memori tersebut tidak memerlukan catu daya cadangan, karena isi memori tidak akan hilang meskipun catu dayanya terputus.

Programmable Read Only Memory (PROM) adalah jenis lain dari memori yang bekerja hampir menyerupai ROM, dengan satu pengecualian yaitu bisa diprogram. PROM dirancang untuk diisi dengan program yang terprogram. Apabila data dapat diubah, maka dapat diadakan pemrograman. Pemrograman ulang dari PROM, membutuhkan perlengkapan khusus yaitu *PROM Programmer*.

b. Perangkat dan *input module*

Perangkat *input* merupakan perangkat keras yang dapat digunakan untuk memberikan sinyal kepada modul *input*. Sistem PLC dapat memiliki jumlah perangkat *input* sesuai dengan sistem yang diinginkan. Fungsi dari perangkat *input* ini yaitu untuk memberikan perintah khusus sesuai dengan kinerja perangkat masukan yang digunakan, misalnya menjalankan atau menghentikan motor. Dalam hal tersebut seperti misalnya, perangkat masukan yang digunakan adalah *push button* yang bekerja secara *Normally Open* (NO) ataupun *Normally Close* (NC).

Terdapat bermacam-macam perangkat *input* yang dapat digunakan dalam pembentukan suatu sistem kontrol seperti misalnya : *selector switches*, *foot switches*, *flow switches*, *sensors* dan lain-lain seperti dalam Gambar 2.5.



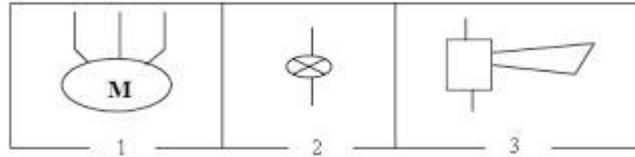
Gambar 2.5 Simbol perangkat *input* PLC
(Sumber : Prayogo, R., 2012)

Keterangan :

1. NO *Push Button*
2. NC *Push Button*
3. NO *Flow Switch*
4. NO *Pressure Switch*

c. Perangkat dan modul *output*

Perangkat *output* adalah komponen-komponen yang memerlukan sinyal untuk mengaktifkan komponen tersebut. Pada sistem PLC dapat mempunyai beberapa perangkat *output* seperti motor listrik, lampu indikator, sirine dan lain-lain seperti dalam Gambar 2.6.



Gambar 2.6 Simbol perangkat *output* PLC
(Sumber : Prayogo, R., 2012)

Keterangan:

1. Simbol Motor Listrik
2. Simbol Lampu
3. Simbol Sirine

d. Catu Daya

Sistem PLC memiliki dua macam catu daya dibedakan berdasarkan fungsi dan operasinya yaitu catu daya dalam dan catu daya luar. Catu daya dalam merupakan bagian dari unit PLC itu sendiri, sedangkan catu daya luar yang memberikan catu daya pada keseluruhan bagian dari sistem termasuk di dalamnya. Catu daya dalam akan mengaktifkan proses kerja pada PLC, dengan besarnya tegangan catu daya yang dipakai disesuaikan dengan karakteristik PLC. Bagian catu daya dalam pada PLC sama dengan bagian-bagian yang lain dimana terdapat langsung pada satu unit PLC atau terpisah dengan bagian yang lain.

e. Pemrograman PLC

Pemrograman PLC adalah memasukkan instruksi-instruksi dasar PLC yang telah membentuk logika pengontrolan yang diinginkan. Bahasa pemrograman biasanya telah disesuaikan dengan ketentuan dari pembuat PLC itu sendiri. Program yang digunakan dalam pemrograman PLC tergantung dari jenis atau merek PLC itu sendiri, jika PLC yang akan dijadikan sebagai bahan penelitian menggunakan PLC merek Omron, maka program yang digunakan adalah *CX-Programmer*.

Program yang akan dimasukkan ke dalam PLC sebagai perintah adalah menggunakan Diagram Tangga (*Ladder Diagram*). *Ladder logic* adalah bahasa pemrograman dengan bahasa grafik atau bahasa yang digambar secara grafik. Pemrogram dengan mudah menggambar skematik diagram dari program pada layar. Hal tersebut menyerupai diagram dasar yang digunakan pada logika sistem kontrol panel, dengan ketentuan instruksi terdiri dari koil - koil, NO, NC dan dalam bentuk penyimbolan.

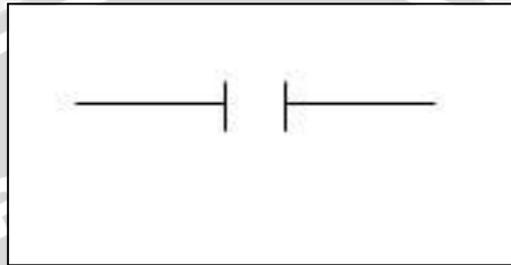
Semua instruksi (perintah pemrograman) merupakan instruksi paling dasar pada PLC. Menurut aturan pemrograman harus ada instruksi dasar END yang oleh PLC dianggap sebagai batas akhir dari program. Instruksi ini tidak ditampilkan pada tombol operasional

programming console, maka pada layar *programming console* akan menampilkan END.

Intruksi – intruksi yang dimaksudkan yaitu :

a. *Load*

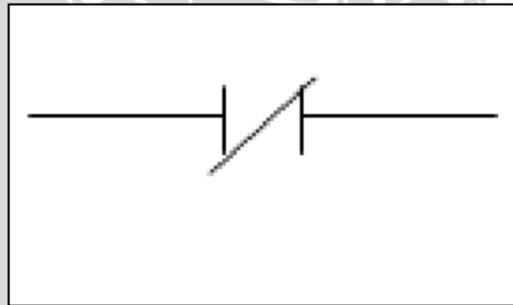
Instruksi ini dibutuhkan jika urutan kerja (*sequence*) pada saat pengontrolan tersebut hanya membutuhkan satu kondisi *logic* saja untuk mengeluarkan satu *output*, dengan logikanya seperti kontak *switch normally open* (Budiyanto, M., 2011) seperti yang disimbolkan dalam Gambar 2.7.



Gambar 2.7 Simbol *load*
(Sumber : Budiyanto, M., 2011)

b. *Load NOT*

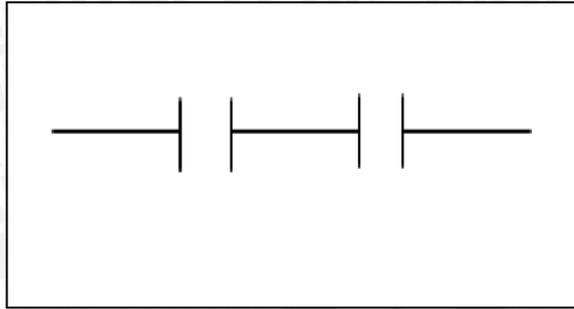
Instruksi ini dibutuhkan jika urutan kerja (*sequence*) pada suatu sistem kontrol, membutuhkan satu kondisi *logic* saja untuk mengeluarkan satu *output*, dengan logikanya seperti kontak *switch normally close* (Budiyanto, M., 2011) seperti yang disimbolkan dalam Gambar 2.8.



Gambar 2.8 Simbol *load not*
(Sumber : Budiyanto, M., 2011)

c. *AND*

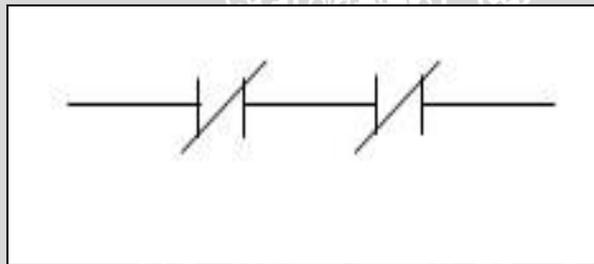
Instruksi ini dibutuhkan jika urutan kerja (*sequence*) pada suatu sistem kontrol, membutuhkan lebih dari satu kondisi *logic* yang harus terpenuhi semuanya untuk mengeluarkan *output*, dengan logikanya seperti dua kontak *switch normally open* yang dipasang seri (Budiyanto, M., 2011) seperti yang disimbolkan dalam Gambar 2.9.



Gambar 2.9 Simbol *And*
(Sumber : Budiyanto, M., 2011)

d. *AND NOT*

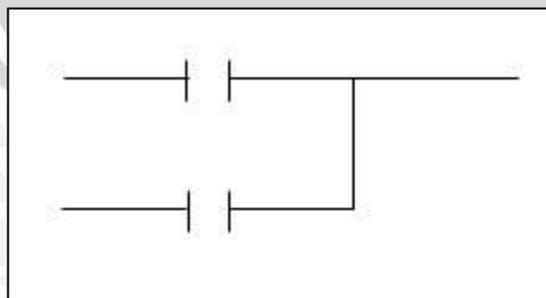
Instruksi ini dibutuhkan jika urutan kerja (*sequence*) pada suatu sistem kontrol, tidak membutuhkan lebih dari satu kondisi logic yang tidak terpenuhi semuanya untuk mengeluarkan satu *output*, dengan logikanya seperti dua kontak *switch normally close* yang dipasang seri (Budiyanto, M, 2011) seperti yang disimbolkan dalam Gambar 2.10.



Gambar 2.10 Simbol *And Not*
(Sumber : Budiyanto, M., 2011)

e. *OR*

Instruksi ini dibutuhkan jika urutan kerja (*sequence*) pada suatu sistem kontrol yang hanya membutuhkan salah satu dari beberapa kondisi *logic* untuk mengeluarkan satu *output*, dengan logikanya seperti dua kontak *switch normally open* yang dipasang paralel (Budiyanto, M., 2011) seperti yang disimbolkan dalam Gambar 2.11.

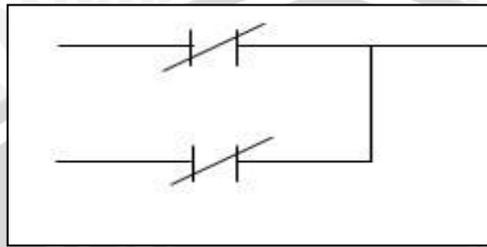


Gambar 2.11 Simbol *Or*
(Sumber : Budiyanto, M., 2011)

f. *OR NOT*

Instruksi ini dibutuhkan jika urutan kerja (*sequence*) pada suatu sistem kontrol tidak hanya membutuhkan salah satu saja, tetapi dari beberapa kondisi *logic* untuk mengeluarkan satu *output*, dengan logikanya seperti dua kontak *switch normally open* yang dipasang parallel relay (Budiyanto, M., 2011) seperti yang disimbolkan dalam Gambar 2.12.

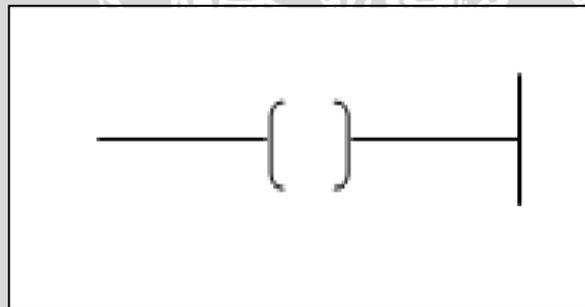
Gambar 2.12 Simbol *Or Not*



(Sumber : Budiyanto, M., 2011)

g. *OUT*

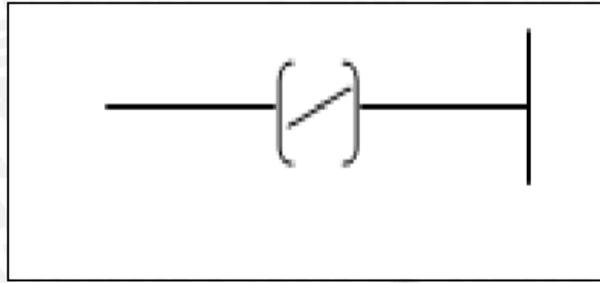
Instruksi ini berfungsi untuk mengeluarkan *output* jika semua logika *ladder diagram* sudah terpenuhi, maka dalam ruang terdapat satu *output* dan *energize* jika mendapatkan input signal diskrit berlogika 1, dengan logikanya seperti coil yang mempunyai multi contactor seperti yang disimbolkan dalam Gambar 2.13.



Gambar 2.13 Simbol *Out*
(Sumber : Budiyanto, M., 2011)

h. *OUT NOT*

Instruksi ini berfungsi untuk mengeluarkan *output* jika semua kondisi logika *ladder diagram* sudah terpenuhi *logic* = 1, maka akan merilis kontak-kontak *switch*-nya, dengan logikanya seperti *coil* yang mempunyai *multi contactor* seperti yang disimbolkan pada Gambar 2.14.



Gambar 2.14 Simbol *Out Not*
(Sumber : Budiyanto, M., 2011)

2.3 PLC OMRON tipe CP1L-L20DTI-D

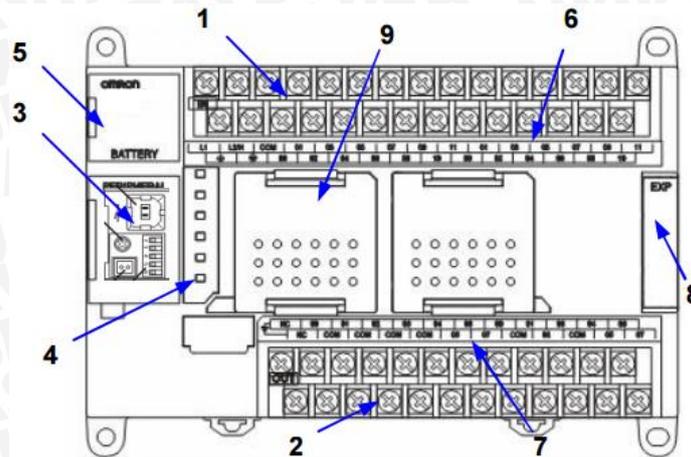
PLC OMRON CP1L adalah salah satu produk PLC dari Omron yang terbaru. CP1L merupakan PLC tipe paket yang tersedia dengan 10,14, 20, 30, 40 atau 60 buah I/O (*input/output*). Sistem *input/output*-nya berupa bit, tetapi lebih dikenal dengan PLC tipe relay karena hanya membaca *input* dan menghasilkan *output* dengan logika 1 atau 0 (Ardiansyah, H., 2013). PLC Omron tipe CP1L-LD20TI-D ditunjukkan dalam Gambar 2.15.



Gambar 2.15 PLC Omron CP1L-L20Dxx

2.3.1 Bagian-bagian umum PLC OMRON CP1L

Seperti dalam Gambar 2.16 PLC OMRON CP1L memiliki bagian – bagian penting diantaranya :

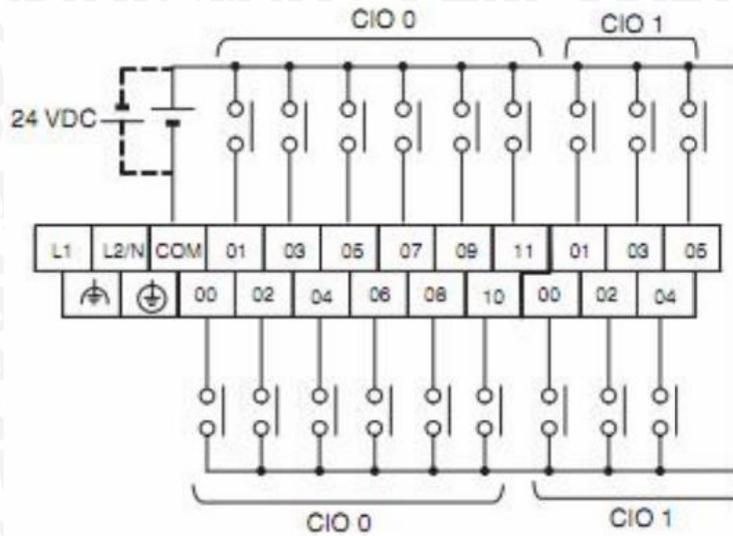


Gambar 2.16 Bagian PLC Omron CP1L
(Sumber : Ardiansyah, H., 2013)

1. Blok *power supply*, *ground* dan *input* terminal.
2. Blok eksternal *power supply* dan *output* terminal.
3. Blok *USB Port* untuk menghubungkan dengan komputer, penggunaan komputer yaitu untuk memprogram dan memonitoring.
4. *Operation indicator* digunakan untuk mengindikasikan status operasi dari CP1L termasuk status *power*, mode operasi, *error*, dan komunikasi *USB*.
5. Baterai untuk mempertahankan *internal clock* dan isi *RAM* ketika suplai *off*.
6. *Input Indicator* berguna untuk menampilkan kondisi menyalnya lampu, jika kontak terminal *input* dalam kondisi menyala.
7. *Output Indicator* berguna untuk menampilkan kondisi menyalnya lampu, jika kontak terminal *output* dalam kondisi menyala.
8. *Expansion I/O unit connector* digunakan untuk menambah *input/output* pada PLC.
9. *Option board slot* digunakan untuk menginstal *RS-232C 1*

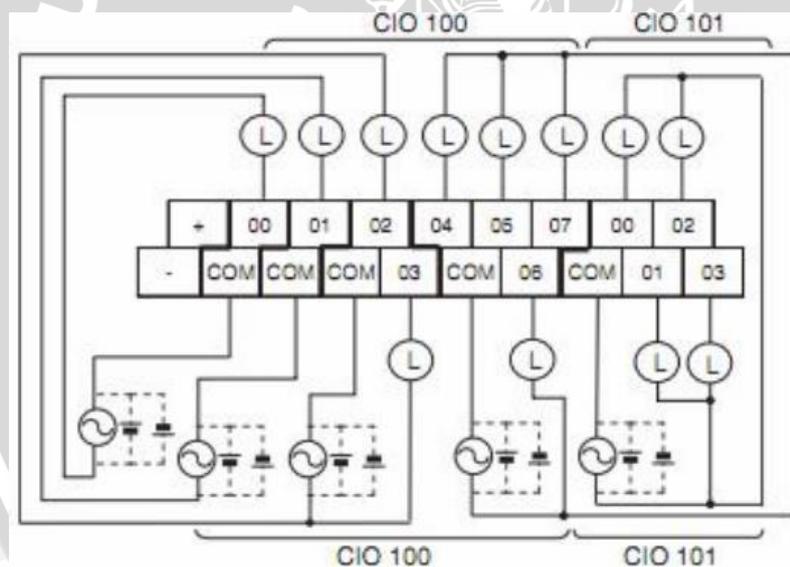
2.3.2 Port terminal Input Output PLC Omron CP1L

Port pada PLC CP1L terdiri dari 12 buah terminal *input* yaitu dari CIO 0.00 – 0.11. Untuk port *output*-nya terdapat 8 buah terminal yaitu dari CIO 100.00 – 100.07. Pada port *input* terdapat dua buah terminal untuk *power supply input* AC PLC, yaitu pada terminal L1 dan L2/N. Port *input* terhubung pada satu titik *common* (COM) (Ardiansyah, H., 2013). *Input* pada terminal COM dapat berupa polaritas + atau negatif – seperti dalam Gambar 2.17.



Gambar 2.17 Port input AC dan DC
(Sumber : Ardiansyah, H., 2013)

Pada *port output* terdapat 4 buah titik COM. Masing masing titik COM terhubung dengan titik *output* yang dibatasi dengan garis batas seperti yang terlihat dalam Gambar 2.18.



Gambar 2.18 Port output AC dan DC
(Sumber : Ardiansyah, H., 2013)

Pada model *power supply* AC terdapat *output* 24 V DC pada terminal *positive* (+) dan *negative* (-).

2.4 Power Supply

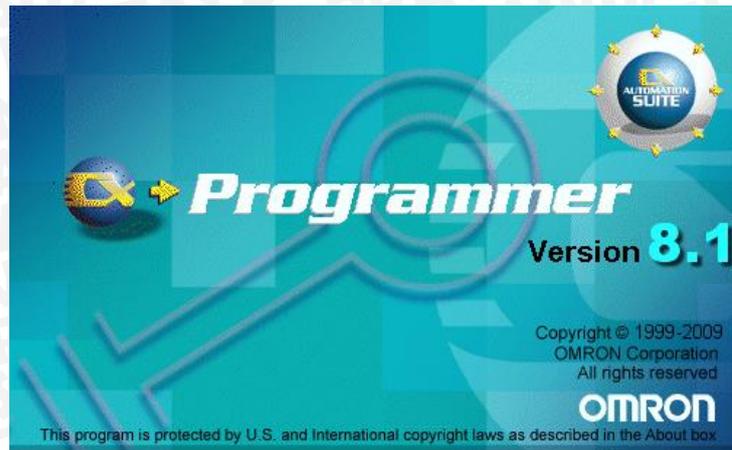
Peralatan - peralatan elektronika tidak akan dapat berfungsi tanpa diberikan masukan berupa tegangan. Dari sekian banyak peralatan elektronika yang telah ada, sebagian besar membutuhkan masukan berupa tegangan searah dalam pengoperasiannya. Tegangan searah ini diperoleh dari suatu alat yang disebut *power supply*. Prinsip kerja *power supply* DC adalah tegangan yang diperoleh akan diteruskan ke logika penurun tegangan daya besar, kemudian diturunkan tegangannya menjadi tegangan yang dibutuhkan untuk sistem berikutnya. Setelah diturunkan tegangan akan masuk ke penyearah yang akan menyearahkan tegangan AC menjadi tegangan DC (Syaprudin, 2010). Berikut ini adalah gambar *power supply* 24 v yang dapat dilihat dalam Gambar 2.19.



Gambar 2.19 *power supply* 24 v
(Sumber : Syaprudin, 2010)

2.5 Program CX-ONE

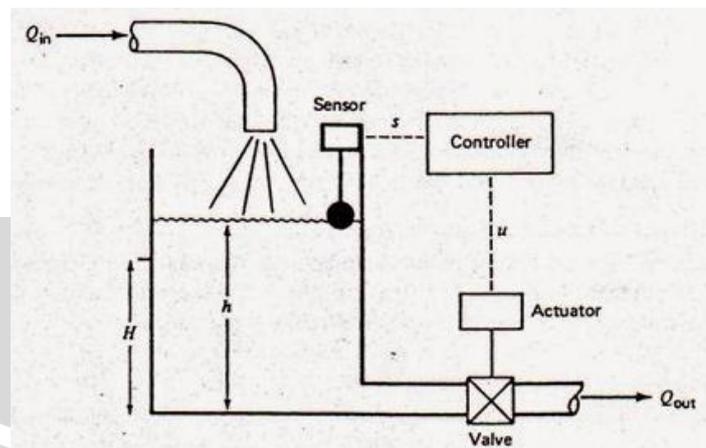
Program *CX-ONE* merupakan perangkat lunak yang dikembangkan untuk memprogram PLC *OMRON* tipe apapun. Program ini dapat membantu memprogram dalam menuliskan program PLC atau memelihara suatu program PLC, serta untuk melakukan pengecekan proses suatu program PLC. Perangkat keras PLC dihubungkan ke PC oleh kabel ethernet. *CX-ONE* memiliki fitur simulasi ladder diagram, dengan demikian ladder diagram yang sudah dibuat pada program tersebut dapat langsung dicoba tanpa perlu menyambungkan PLC nyata dengan PC. Program yang dipakai dapat dilihat dalam Gambar 2.20.



Gambar 2.20 Program CX-One.
(Sumber : Arifin, M., 2016)

2.6 Sistem Kontrol Otomasi

Sistem kontrol otomasi adalah suatu sistem pengendalian dimana variabel manipulator dan variabel kontrol bekerja dengan sistem yang dilakukan oleh sebuah peralatan pengontrol otomatis, baik dari segi pengamatan *input* pengolahan data serta menggerakkan peralatan *output* (Wicaksono, H., 2009). Contoh sistem kontrol otomatis yaitu pada sebuah tangki air, dimana *controller* akan otomatis menggerakkan actuator ketika ketinggian air menyentuh sensor, sehingga keran pengeluaran terbuka. Kejadian ini terus terjadi secara berulang dan kontinu seperti dalam Gambar 2.21.



Gambar 2.21 Pengisian air dengan pengontrolan otomatis.
(Sumber : Wicaksono, H., 2009)

2.7 Sistem Kontrol Sekuensial

Kontrol sekuensial (kontrol berurutan) adalah teknik pengontrolan yang digunakan untuk mengatur suatu operasi yang saling terkait, terhubung atau terencana (terjadwal). Terdapat tiga kategori kontrol sekuensial yaitu sistem melaksanakan urutan berikutnya jika kondisi yang ditentukan sebelumnya terpenuhi (*conditional control*), sistem melaksanakan urutan berikutnya jika telah mencapai waktu yang telah ditentukan (*time schedule control*), dan sistem dimana waktu pelaksanaan atau interval waktu tidak penting, hanya urutan operasi yang telah ditetapkan yang dipentingkan (*executive control*) (Syaprudin, 2010).

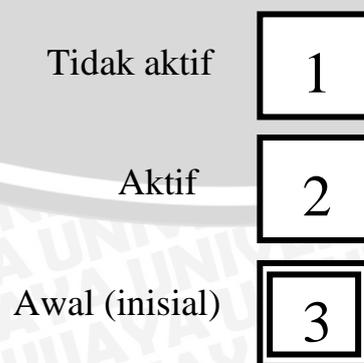
Rangkaian kontrol sekuensial dapat dengan mudah dirancang dalam bentuk peralatan yang disebut kontroler sekuensial. Kontroler ini menggunakan komputer khusus yang dirancang untuk kebutuhan kontrol sekuensial dan dapat melaksanakan perintah sekuensial untuk berbagai penggunaan.

2.8 Grafcet

Grafcet merupakan grafik yang terdiri dari dua simbol, yaitu simbol langkah (*step*) dan simbol transisi (*transition*). Suatu grafik *grafcet* minimal memiliki sebuah langkah dan sebuah transisi. Panah lurus terhubung satu sama lain dari sebuah langkah ke sebuah transisi atau sebuah transisi ke sebuah langkah.

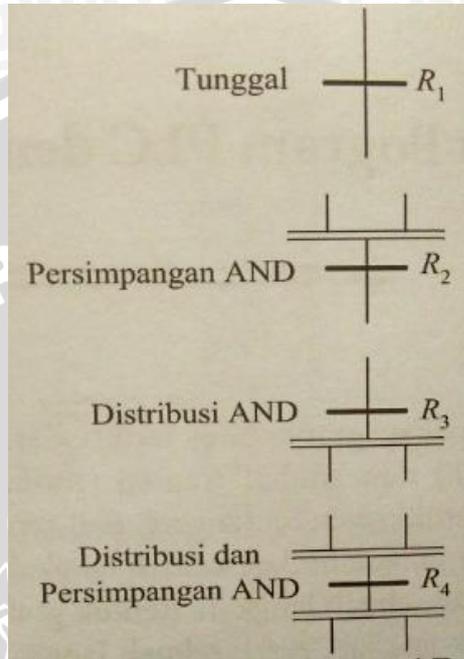
2.8.1 Langkah dan Transisi

Suatu langkah disimbolkan seperti bujur sangkar. Bagian langkah dimungkinkan memiliki dua keadaan aktif, keadaan ini diwujudkan dengan sebuah tanda di dalam langkah seperti bentuk langkah 2 dan keadaan tidak aktif seperti bentuk langkah 1. Langkah-langkah yang harus aktif saat sistem dijalankan harus disimbolkan dengan bujur sangkar ganda, seperti dalam Gambar 2.22.



Gambar 2.22 Bentuk gambar transisi aktif dan tidak aktif.
(Sumber : Rusli, M., 2012)

Sebuah transisi disimbolkan seperti garis vertikal tanpa tanda panah, dengan urutan berasal dari puncak menuju ke bagian dasar. Simbol transisi adalah sebuah batang, tapi pada beberapa keadaan bagian akhirnya harus beralih dan diikuti oleh batang ganda, seperti dalam Gambar 2.23.

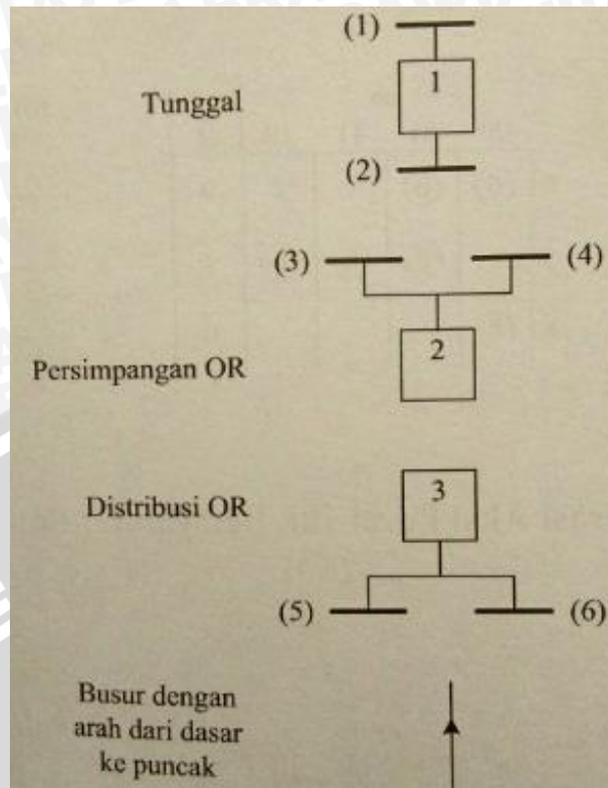


Gambar 2.23 Bentuk gambar transisi.
(Sumber : Rusli, M., 2012)

Perubahan menjadi batang ganda terjadi ketika dua buah busur atau lebih bergabung menjadi transisi 2 dan 4. Hal ini menyatakan bahwa diperlukan proses untuk menunggu, agar keadaan masukan diaktifkan terlebih dahulu sebelum transisi diaktifkan. Sebuah transisi diikuti oleh batang ganda ketika kedua busur atau lebih meninggalkan transisi 3 dan 4.

Penerimaan R_i berhubungan dengan setiap transisi. Penerimaan adalah sebuah fungsi variabel *input grafcet*. Bentuk hubungan haruslah mengalir dari langkah ke transisi atau dari transisi ke langkah.

Saat dua hubungan atau lebih meninggalkan langkah yang sama, mereka mempunyai ujung keberangkatan yang sama. Busur tegak mengalir dari dasar ke puncak yang harus ditandai dengan tanda panah, seperti dalam Gambar 2.24.



Gambar 2.24 Bentuk hubungan.
(Sumber : Rusli, M., 2012)

Sebuah transisi tanpa langkah *input* diketahui sebagai sumber transisi dan sebuah transisi tanpa langkah *output* diketahui sebagai *input* transisi.

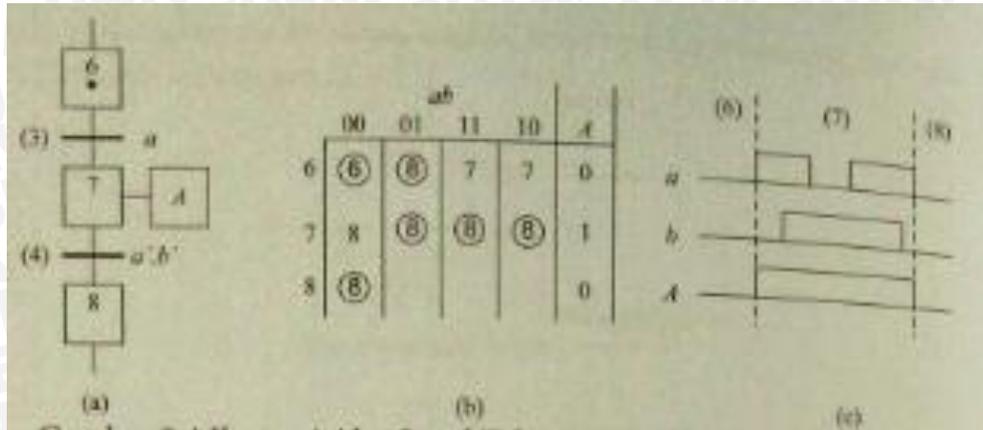
2.8.2 Pengaktifan transisi

Langkah aktif hanya mengandung satu tanda yaitu langkah yang tidak aktif dan tidak memiliki tanda apapun. Semua langkah-langkah aktif yang diberi tanda pada saat tertentu merupakan penjelasan situasi keadaan pada saat itu. Situasi dapat disamakan dengan keadaan dari suatu sistem, perubahan situasi juga dapat dicapai dengan mengaktifkan transisi. Masukan pengendali logika saling terhubung dengan transisi-transisi yang membangkitkan pengaktifan, sedangkan *output*-nya terhubung dengan langkah-langkah yang dihasilkan oleh tempat yang diaktifkan. Transisi dapat diaktifkan jika kedua syarat berikut dipenuhi yaitu semua langkah yang mendahului transisi telah aktif dan penerimaan transisi adalah benar.

2.8.3 Aksi dan *Output*

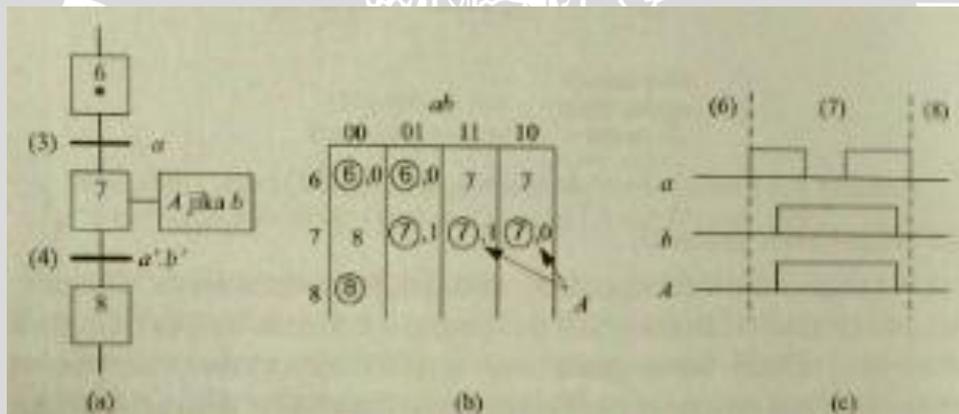
Terdapat dua kategori aksi, yaitu aksi impuls dan aksi level. Sebuah aksi level dimodelkan oleh variable Boolean. Hal ini bisa bersyarat atau tidak bersyarat. Aksi A adalah yang tidak bersyarat, hal ini berarti bahwa aksi A dianggap bernilai satu pada

saat langkah ketujuh aktif. Pada saat langkah ini aktif tanpa kondisi yang lain, hal ini digambarkan dengan tabel keadaan seperti dalam Gambar 2.25.



Gambar 2.25 Instruksi level tak bersyarat.
(Sumber : Rusli, M., 2012)

Sedangkan aksi A adalah aksi bersyarat jika aksi A dianggap bernilai satu dan b bernilai satu pada saat langkah ketujuh aktif, seperti dalam Gambar 2.26.

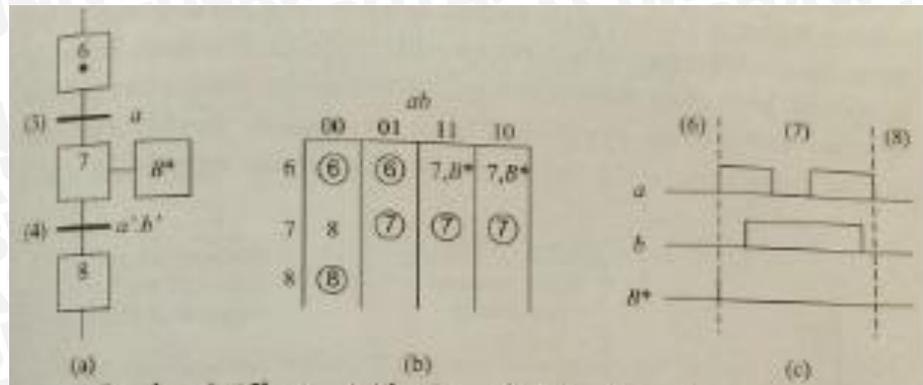


Gambar 2.26 Instruksi level bersyarat.
(Sumber : Rusli, M., 2012)

Aksi impuls dapat dipertanggungjawabkan untuk mengubah nilai sebuah variabel diskrit. Pada akhirnya dapat berupa variabel Boolean atau variabel diskrit yang lain (contoh : nilai suatu waktu / timer). Sebuah aksi impuls yang berhubungan dengan sebuah langkah kemudian diteruskan dengan segera oleh langkah tersebut dan dapat mengubah keadaan dari aktif menjadi tidak aktif, tanpa menghiraukan waktu selama langkah dalam keadaan tetap aktif.

Aksi B' adalah sebuah aksi impuls yaitu tanda bintang menandakan sifat impuls adalah sebuah aksi. Sebagai contoh, penambahan sebuah delay (Timer) yaitu T dapat dituliskan sebagai $(T \rightarrow T + 1)^*$ atau secara sederhana $(T \rightarrow T + 1)$

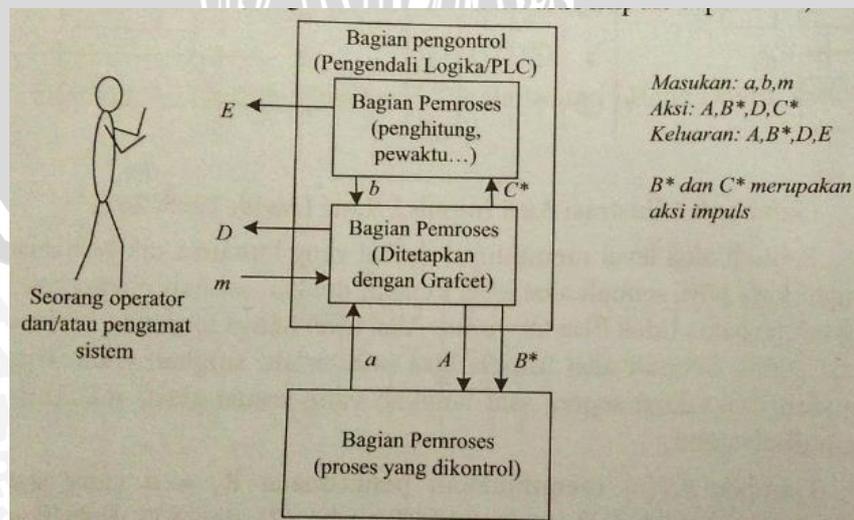
selama tidak ada arti ganda pada sifat impuls aksi tersebut seperti dalam Gambar 2.27.



Gambar 2.27 Ilustrasi aksi impuls.
(Sumber : Rusli, M., 2012)

Sebuah aksi level mempunyai durasi yang lamanya dan tak terbatas. Dengan kata lain, sebuah aksi level mempunyai sebuah durasi yang singkat dan tak terbatas. Aksi level hanya terdefiniskan untuk situasi stabil, sebuah aksi impuls bisa saja terlalu singkat jika aksi yang dieksekusi disegerakan pada saat langkah yang sesuai telah aktif, walaupun sistem belum stabil.

Bagian *output* yang identik dengan sebuah sinyal yang beraksi pada lingkungan dalam suatu sistem. Hal ini ditunjukkan aksi impuls C^* . Aksi tersebut bukanlah sebuah *output* tetapi mengontrol sebuah operasi bagian proses pada kontrol logikanya. Variabel E adalah *output* pengendalian logika tetapi bukan merupakan aksi didalam pengertian *grafcet*, seperti dalam Gambar 2.28.

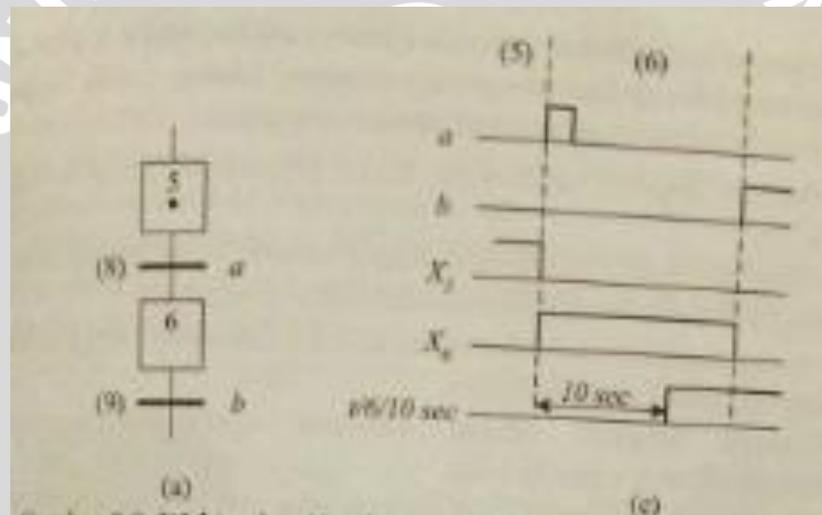


Gambar 2.28 Pengendalian logika, aksi, input dan output.
(Sumber : Rusli, M., 2012)

2.8.4 Keadaan Internal dan Waktu

Variabel X_i adalah variabel Boolean yang bernilai 1 saat langkah ke-1 aktif yaitu $X_5 = 1$ dan $X_6 = 0$. Variabel $t/i/D$ adalah variabel Boolean yang sama dengan 1 jika waktu akhirnya sama dengan nilai D dan telah melewati sejak langkah i berubah dari tidak aktif menjadi aktif. Hal ini diilustrasikan dalam langkah 5. Langkah 5 aktif, yaitu $X_5 = 1$ pada saat variabel a dianggap bernilai 1. Variabel Boolean $t/6/10$ detik bernilai 1 saat sepuluh detik kemudian, keadaan ini dipertahankan sampai variabel $X_6 = 0$ hingga berubah dari 0 ke 1. Jika pada saat langkah 6 tidak aktif, maka tidak berpengaruh.

Variabel X_i atau $t/i/D$ dapat digunakan dalam sebuah penerimaan atau sebagai kondisi untuk aksi level, jika Sebuah peristiwa ($t/i/D$) dapat digunakan untuk sebuah penerimaan seperti dalam Gambar 2.29.

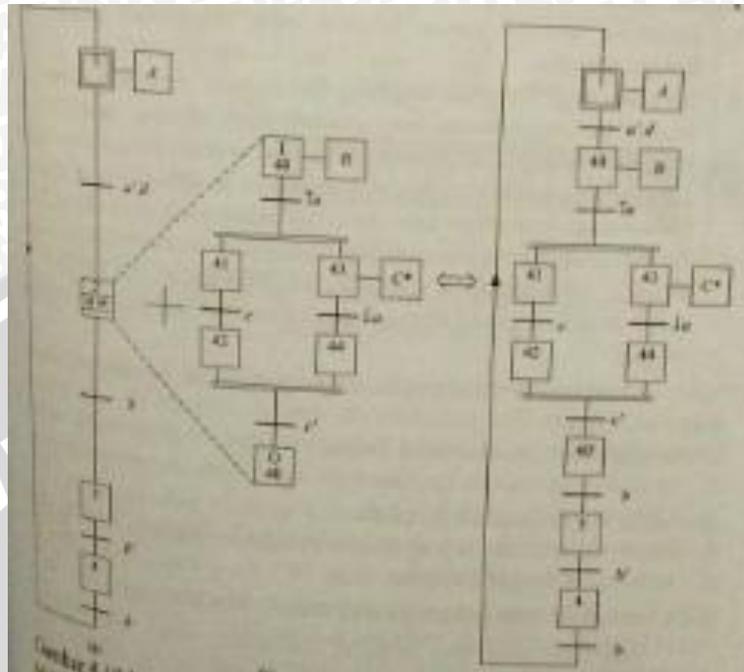


Gambar 2.29 Waktu dan Keadaan Internal.
(Sumber : Rusli, M., 2012)

2.8.5 Makrostep

Makrostep digunakan untuk mempersingkat grafik pada sebuah *grafcet*. Tujuan konsep makrostep adalah untuk memudahkan deskripsi sistem yang kompleks. Makrostep memberikan kemudahan pengertian tentang bentuk grafik sebuah *grafcet*. Sebuah ekspansi makrostep harus memenuhi syarat yaitu sebuah makrostep hanya mempunyai sebuah *input* dan sebuah *output*, semua pengaktifan sebuah transisi ke hulu makrostep dan mengaktifkan langkah *input* ekspansinya, *output* langkah sebuah ekspansi makrostep berperan

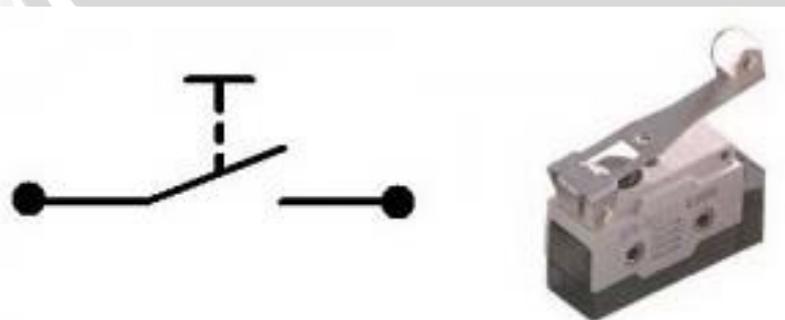
mengaktifkan transisi ke hilir, dan tidak ada hubungan yang meninggalkan ekspansi makrostep, seperti dalam Gambar 2.30.



Gambar 2.30 Bentuk gambar makrostep.
(Sumber : Rusli, M., 2012)

2.9 Sensor *Limit Switch*

Limit switch merupakan jenis sakelar yang dilengkapi dengan katup yang berfungsi menggantikan tombol. Prinsip kerja *limit switch* sama seperti sakelar *Push on* yaitu hanya akan terhubung pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan, kemudian akan memutuskan pada saat katupnya tidak ditekan (Yusran, A., 2013). *Limit switch* termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanis pada sensor tersebut. Penerapan dari *limit switch* adalah sebagai sensor posisi suatu benda yang bergerak. Simbol *limit switch* ditunjukkan dalam Gambar 2.31.

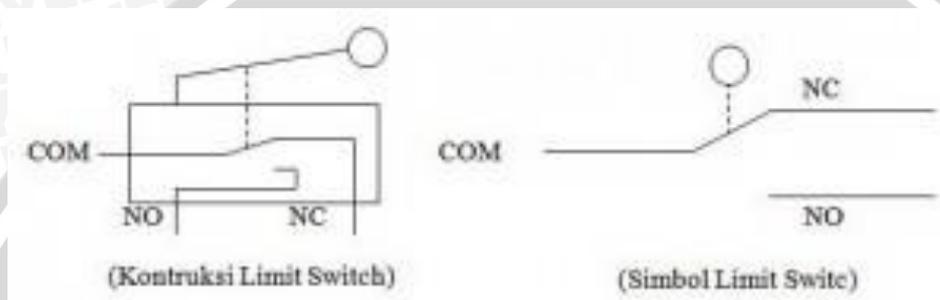


Gambar 2.31 *Limit switch*
(Sumber : Yusran, A., 2013)

Limit switch umumnya digunakan untuk :

1. Memutuskan dan menghubungkan rangkaian menggunakan objek atau benda lain.
2. Menghidupkan daya yang besar dengan sarana yang kecil.
3. Sebagai sensor posisi atau kondisi suatu objek.

Prinsip kerja *limit switch* diaktifkan dengan penekanan pada tombolnya pada batas yang telah ditentukan sebelumnya sehingga terjadi pemutusan atau penghubungan rangkaian dari rangkaian tersebut. *Limit switch* memiliki 2 kontak yaitu *Normally Open* (NO) dan kontak *Normally Close* (NC) dimana salah satu kontak akan aktif jika tombolnya tertekan. Konstruksi dan simbol *limit switch* dapat dilihat seperti dalam Gambar 2.32.

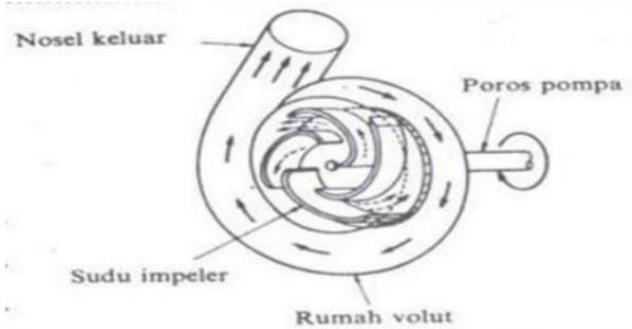


Gambar 2.32 Bagian NC, No, dan COM pada *limit switch*
(Sumber : Yusran, A., 2013)

2.10 Pompa Sentrifugal

Pompa sentrifugal digunakan untuk memberikan atau menambah kecepatan pada cairan, kemudian merubahnya menjadi energi tekanan. Cairan dipaksa masuk ke sebuah *impeller*, dengan daya dari luar diberikan kepada poros pompa untuk memutar *impeller* yang ada berada dalam cairan tadi. Apabila *impeller* berputar, maka zat cair yang ada dalam *impeller* akan ikut berputar akibat dorongan sudu – sudu pada *impeller*. Karena timbul gaya sentrifugal maka zat cair mengalir dari tengah *impeller* menuju keluar melalui saluran diantara sudu – sudu dengan kecepatan tinggi (Faisal, M., 2016).

Zat cair yang meninggalkan *impeller* tersebut dikumpulkan di dalam rumah pompa yang berbentuk spiral atau biasanya disebut volut yang tugasnya mengumpulkan cairan dari *impeller* dan mengarahkan ke *discharge nozzle* (Faisal, M, 2016). *Discharge nozzle* berbentuk seperti kerucut sehingga kecepatan aliran yang tinggi dari *impeller* bertahap turun, bentuk kerucut ini disebut *diffuser*. Pada waktu penurunan kecepatan di dalam *diffuser* diubah menjadi energi tekanan. Jadi *impeller* pompa berfungsi memberikan kerja pada zat cair sehingga energi yang dikandungnya akan menjadi lebih besar (Faisal, M., 2016), seperti yang ditunjukkan dalam Gambar 2.33.



Gambar 2.33 Pompa Sentrifugal
(Sumber : Faisal, M., 2016)

Bagian – bagian utama pompa sentrifugal antara lain :

a. Impeller

Impeller berfungsi untuk mengubah energi mekanis dari pompa menjadi energi kecepatan pada cairan yang dipompa secara kontinue, sehingga cairan pada sisi hisap secara terus menerus akan masuk mengisi kekosongan akibat perpindahan dari cairan yang masuk sebelumnya. Ada tiga jenis *impeller*, yaitu *open impeller*, *semi open impeller* dan *closed impeller* (Faisal, M., 2016) seperti dalam Gambar 2.34.



Gambar 2.34 Macam – macam *impeller*
(Sumber : Faisal, M., 2016)

Pompa sentrifugal dapat menggunakan dua macam *impeller*, yaitu isapan tunggal dan isapan ganda. Pada pompa sentrifugal di *Waste Water Treatment* menggunakan pompa isapan tunggal.

b. Rumah Pompa (Volute Casing)

Rumah pompa merupakan bagian paling luar dari pompa yang berfungsi sebagai pelindung elemen yang berputar, tempat kedudukan *diffuser*, *suction nozzle*, dan *discharge nozzle* serta memberikan arah aliran dari *impeller* dan mengubah energi kecepatan menjadi energi tekan (Faisal, M., 2016) seperti dalam Gambar 2.35.



Gambar 2.35 *Volute casing*
(Sumber : Faisal, M., 2016)

c. Shaft

Shaft berfungsi untuk meneruskan putaran dari penggerak (motor) selama beroperasi ke *impeller*. *Shaft* juga berfungsi sebagai tempat kedudukan *impeller* dan bagian – bagian lain yang berputar. Untuk menghubungkan antara *shaft* pompa dengan *shaft* penggerak motor maka diperlukan koping. Bagian luar *shaft* ini biasanya dilindungi oleh *shaft sleeve* (Faisal, M., 2016) seperti dalam Gambar 2.36.



Gambar 2.36 *Shaft* pada pompa air
(Sumber : Faisal, M., 2016)

d. Shaft Sleeve

Shaft sleeve berbentuk silinder berlubang yang berfungsi untuk melindungi *shaft* utama dari erosi, korosi, dan aus. Apabila *shaft* utama mengalami kerusakan maka *shaft* utama tidak bisa diperbaiki tetapi harus dilakukan penggantian dengan yang baru (Faisal, M., 2016). *Shaft sleeve* ditampilkan dalam Gambar 2.3



Gambar 2.37 *Shaft Sleeve*
(Sumber : Faisal, M., 2016)

e. Gland Packing

Gland packing ini berfungsi untuk mengurangi kebocoran cairan dalam *casing* pompa dan mencegah udara dari luar masuk ke dalam pompa. Apabila ada udara luar yang masuk ke dalam pompa maka akan mengakibatkan performa pompa akan menurun. *Gland packing* ditampilkan dalam Gambar 2.38.



Gambar 2.38 *Gland packing*
(Sumber : Arifin, M., 2016).

f. Stuffing Box

Stuffing box berfungsi untuk mencegah kebocoran pada daerah dimana poros pompa menembus *casing*. Jika pompa bekerja dengan *suction lift* dan tekanan pada ujung *stuffing box* lebih rendah dari tekanan atmosfer, maka *stuffing box* berfungsi untuk mencegah kebocoran udara masuk kedalam pompa. Dan bila tekanan lebih besar daripada tekanan atmosfer, maka berfungsi untuk mencegah kebocoran cairan keluar pompa. *Stuffing box* ditampilkan dalam Gambar 2.39.



Gambar 2.39 *Stuffing box*
(Sumber : Arifin, M., 2016).

g. Bearing

Bearing (bantalan) berfungsi untuk menumpu dan menahan beban dari poros agar dapat berputar, baik berupa beban radial maupun beban aksial. *Bearing* juga memungkinkan poros untuk dapat berputar dengan lancar dan tetap pada tempatnya, sehingga kerugian gesek juga akan kecil. *Bearing* ditampilkan dalam Gambar 2.40.



Gambar 2.40 *Bearing*
(Sumber : Arifin, M., 2016).

h. Oil Seal

Seal ini berfungsi untuk menjaga oli yang berada di dalam *bearing housing* agar tidak bocor. *Oil Seal* ditampilkan dalam Gambar 2.41.



Gambar 2.41 Oil seal
(Sumber : Arifin, M., 2016).

2.11 Smart Pump Plant

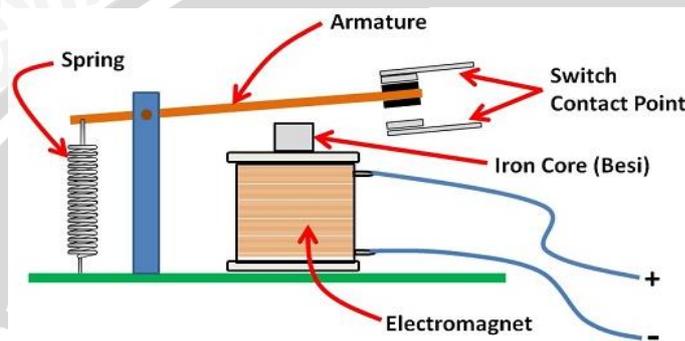
Smart pump plant adalah *plant* berbentuk alat yang terdiri dari 2 wadah plastik, 8 pompa air, 3 sensor LM35, dan 3 sensor *limit switch*. 8 pompa air digunakan sebagai keadan kombinasi, agar pompa tidak hanya memiliki satu *input* untuk menjalankan sebuah pompa air. Pengontrolan sekuensial digunakan untuk memaksimalkan kinerja dari pompa air. wadah dengan kapasitas 3 liter air digunakan untuk menampung air. *Smart pump plant* memiliki tegangan 24 volt yang sama seperti tegangan PLC yaitu 24 volt. Terdapat 3 kondisi *input* untuk mengaktifkan pompa air, yaitu kondisi pertama sensor LM35 yang berfungsi sebagai input pompa M1, M2, dan M3, kondisi kedua yaitu sensor *limit switch* yang berfungsi sebagai input pompa M4, M5, dan M6, dan kondisi ketiga yaitu *timer* yang berfungsi sebagai input pompa M7 dan M8 seperti yang ditampilkan dalam Gambar 2.42.



Gambar 2.42 Smart pump plant

2.12 Relay

Relay adalah sakelar mekanik yang dikendalikan atau dikontrol secara elektronik (elektro magnetik). Sakelar pada *relay* akan terjadi perubahan posisi *off* ke *on* pada saat diberikan energi elektro magnetik pada *armature relay* tersebut. *Relay* pada dasarnya terdiri dari 2 bagian utama yaitu sakelar mekanik dan sistem pembangkit elektromagnetik (induktor inti besi) (Irjan, L., 2013). Sakelar dikendalikan menggunakan tegangan listrik yang diberikan ke induktor pembangkit magnet, kemudian digunakan untuk menarik *armature* tuas saklar atau kontak relay, seperti yang dapat dilihat dalam Gambar 2.43.



Gambar 2.43 Bagian *relay* elektromagnetik
(Sumber : Irjan, L., 2013)

Relay dibutuhkan dalam rangkaian elektronika sebagai eksekutor sekaligus *interface* antara beban dan sistem kendali elektronik yang berbeda sistem *power supply*-nya. Secara fisik antara sakelar dengan elektromagnetik *relay* terpisah sehingga antara beban dan sistem kontrol akan terpisah. Bagian utama *relay* elektro mekanik adalah sebagai berikut :

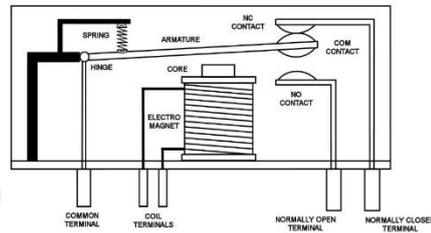
1. Kumparan elektromagnet
2. Sakelar
3. *Armature Spring* (Pegas).

Dari konstruksi *relay* elektro mekanik dapat diuraikan sistem kerja atau proses *relay* bekerja. Pada saat elektromagnet tidak diberikan sumber tegangan, maka tidak ada medan magnet yang menarik *armature*, sehingga saklar relay tetap terhubung ke terminal *Normally Close* (NC). Kemudian pada saat elektromagnet diberikan sumber tegangan, maka terdapat medan magnet yang menarik *armature*, sehingga saklar relay terhubung ke terminal *Normally Open* (NO) (Irjan, L, 2013).

Relay elektro mekanik memiliki kondisi sakelar dalam 3 posisi. Ketiga posisi sakelar *relay* ini akan berubah pada saat *relay* mendapat tegangan sumber pada elektromagnetnya. Ketiga posisi saklar relay tersebut yaitu :

1. *Normaly On*

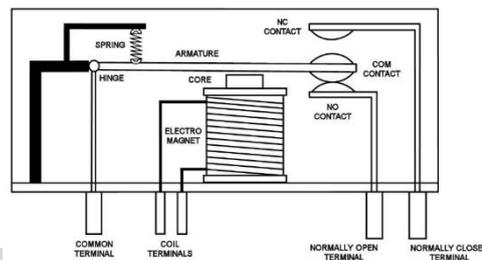
Kondisi awal sakelar tertutup (*On*) dan akan terbuka (*Off*) jika *relay* diaktifkan dengan cara memberi arus yang sesuai pada kumparan atau koil *relay*. Istilah lain kondisi ini adalah *Normaly Close* (NC) seperti dalam Gambar 2.44.



Gambar 2.44 Relay *normaly close* (NC).
(Sumber : Irjan, L., 2013)

2. *Normaly Off*

Kondisi awal sakelar terbuka (*Off*) dan akan tertutup jika *relay* diaktifkan dengan cara memberi arus yang sesuai pada kumparan atau koil *relay*. Istilah lain kondisi ini adalah *Normaly Open* (NO) seperti dalam Gambar 2.45.



Gambar 2.45 Relay *normaly open* (NO).
(Sumber : Irjan, L., 2013)

3. *Change-Over* (CO) atau *Double-Throw* (DT)

Relay jenis ini memiliki dua pasang terminal dengan dua kondisi yaitu *Normaly Open* (NO) dan *Normaly Close* (NC).

Relay juga dibedakan berdasar *pole* dan *throw* yang dimilikinya. *Pole* adalah banyaknya kontak yang dimiliki oleh *relay*. Sedangkan *throw* adalah banyaknya kondisi (*state*) yang mungkin dimiliki sakelar (Effendi, A., 2013). Berikut ini penggolongan *relay* berdasar jumlah *pole* dan *throw* atau disebut juga sebagai simbol *relay*, yaitu :

1. *Single Pole Single Throw (SPST)*

Relay ini memiliki empat terminal yaitu, dua terminal kumparan atau koil dan dua terminal saklar (A dan B) yang dapat terhubung dan terputus seperti dalam Gambar 2.46.



SPST

Gambar 2.46 *Single Pole Single Throw (SPST)*
(Sumber : Effendi, A., 2013)

2. *Single Pole Double Pole (SPDT)*

Relay ini memiliki lima terminal, yaitu dua terminal kumparan atau koil dan tiga terminal saklar (A, B, dan C) yang dapat terhubung dan terputus dengan satu terminal pusat. Jika suatu saat terminal (misal A) terputus dengan terminal pusat (C) maka terminal lain (B) terhubung dengan terminal pusat tersebut (C), demikian juga sebaliknya seperti dalam Gambar 2.47.

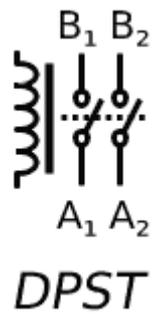


SPDT

Gambar 2.47 *Single Pole Double Pole (SPDT)*
(Sumber : Effendi, A., 2013)

3. *Double Pole Single Throw (DPST)*

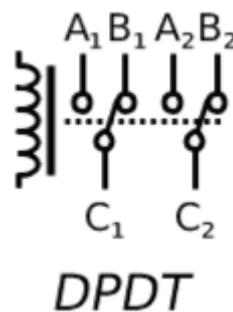
Relay ini mempunyai enam terminal, yaitu dua terminal kumparan atau koil dan empat terminal, merupakan dua pasang sakelar yang dapat terhubung dan terputus (A1 dan B1 serta A2 dan B2) seperti dalam Gambar 2.48.



Gambar 2.48 *Double Pole Single Throw (DPST)*
(Sumber : Effendi, A., 2013)

4. *Double pole Double Throw (DPDT)*

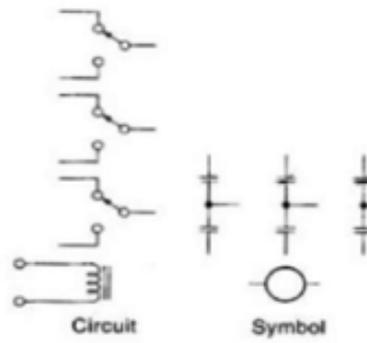
Relay ini mempunyai delapan terminal, yaitu dua terminal kumparan atau koil, enam terminal merupakan dua saklar yang dapat terputus dan terhubung (A_1, B_1, C_1 dan A_2, B_2, C_2) seperti dalam Gambar 2.49.



Gambar 2.49 *Double pole Double Throw (DPDT)*
(Sumber : Effendi, A., 2013)

5. *Quadruple Pole Double Throw (QPDT)*

QPDT sering disebut sebagai *Quad Pole Double Throw* atau *Four Pole Double Throw (4PDT)*. *Relay* ini setara dengan empat buah saklar atau relay SPDT atau dua buah relay DPDT dan terdiri dari empat belas pin (termasuk 2 buah untuk koil) seperti dalam Gambar 2.50.

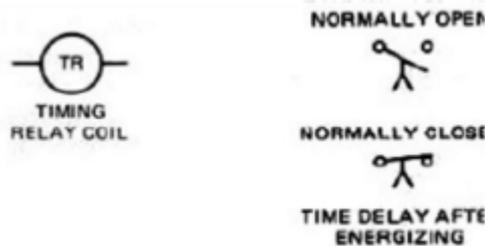


Gambar 2.50 *Quadruple Pole Double Throw (QPDT)*
(Sumber : Effendi, A., 2013)

Selain itu terdapat jenis-jenis *relay* lainnya. *Relay* juga diaplikasikan untuk berbagai keperluan, sehingga dianggap *relay* tersebut memiliki fungsi khusus, seperti contoh berikut :

a. *Timing Relay*

Relay yang bekerja untuk sebuah pewaktuan, dimana koil *relay* akan dianggap *on*, jika memenuhi beberapa waktu tertentu (misal 5 detik). *Timing relay* adalah jenis *relay* yang khusus. Cara kerjanya yaitu jika koil dari *timing relay on*, maka beberapa detik kemudian baru kontak *relay* akan *on* atau *off* (sesuai jenis NO/NC *contact*) seperti dalam Gambar 2.51.

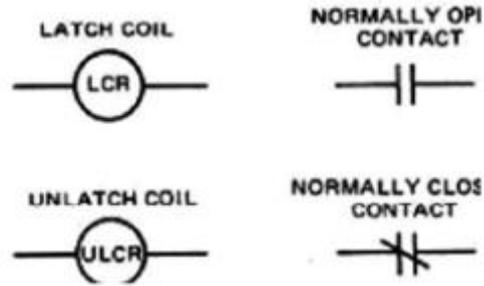


Gambar 2.51 *Timing Relay*

(Sumber : Effendi, A., 2013)

b. *Latching Relay*

Relay ini dipergunakan untuk *latching* atau mempertahankan kondisi aktif *input* sekalipun *input* sebenarnya sudah mati. Cara kerjanya yaitu jika *latch coil* diaktifkan, maka *latch coil* tidak akan bisa dimatikan kecuali *unlatch coil* diaktifkan seperti dalam Gambar 2.52.



Gambar 2.52 Latching Relay

(Sumber : Effendi, A., 2013)

c. *Reed Relay*

Relay ini memiliki seperangkat kontak di dalam vakum atau gas inert untuk mengisi tabung gelas yang melindungi kontak terhadap korosi atmosfer. Kontak tertutup oleh medan magnet yang dihasilkan ketika arus mengalir melalui kumparan di sekeliling tabung gelas. *Reed relay* mampu mengubah kecepatan lebih cepat daripada jenis *relay* yang lebih besar.



BAB III

METODE PENELITIAN

Penyusunan skripsi ini merupakan perencanaan dan pembuatan sistem otomatisasi pada motor pompa air yang dikemas menjadi sebuah alat *Smart Pump Plan* dengan menggunakan PLC. Langkah – langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut :

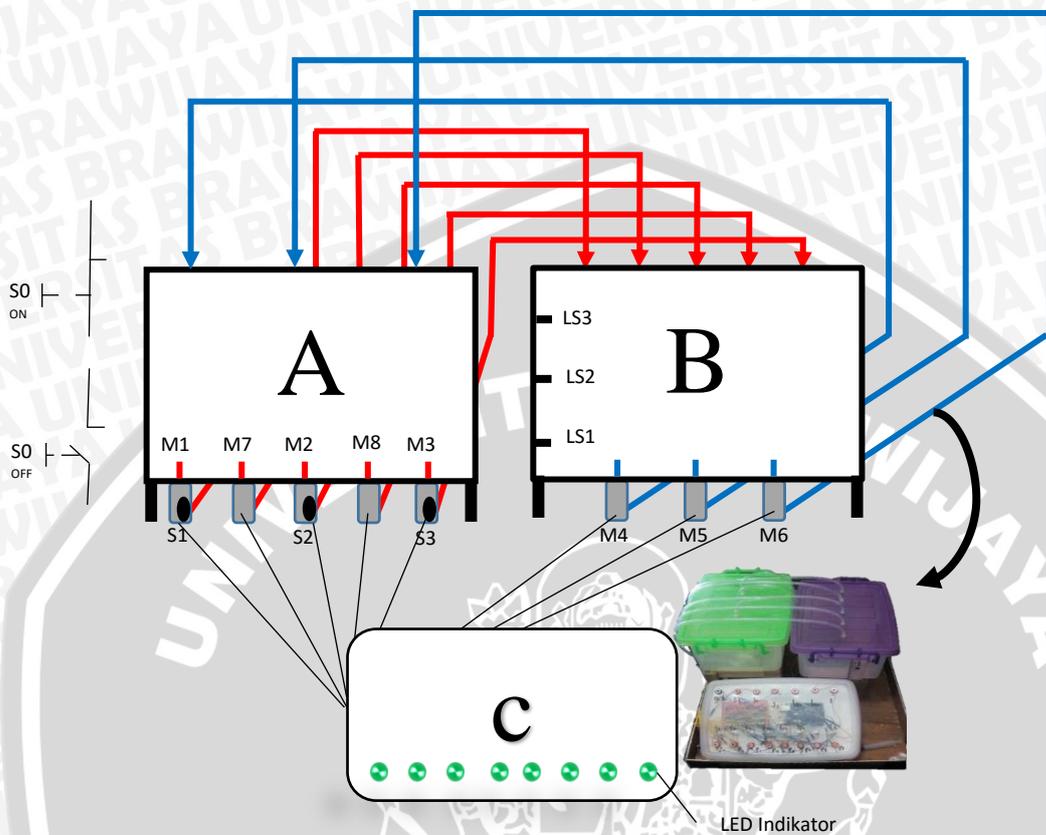
1. Perancangan Perangkat Keras
2. Spesifikasi Alat
3. Perancangan Sistem Kerja
4. *Input* dan *Output*
5. Perancangan Diagram State
6. Perancangan Grafcet
7. Perancangan Ladder Diagram
8. Perancangan Diagram Alir
9. Pengambilan Kesimpulan

3.1. Perancangan Perangkat Keras

Untuk merealisasikan alat atau sistem yang telah dirancang sebelumnya, maka perlu diperhatikan beberapa tahapan dalam pembuatan alat yaitu perancangan alat. Perancangan alat dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem otomatisasi *smart pump plant* tersebut dapat berjalan dengan baik Perancangan sistem yang dilakukan meliputi penjelasan mengenai alat miniatur *smart pump plant*.

Smart pump plant adalah alat yang dibuat dari bahan plastik bening. Bahan tersebut berukuran 17 cm x 23 cm dengan tinggi 18 cm. Pada alat tersebut terdapat 8 pompa yang berfungsi sebagai penyedot air, agar air dapat berpindah dari satu wadah ke wadah lainnya. Dari 8 pompa terdapat 3 kondisi *input* yaitu kondisi pertama 3 pompa dipicu oleh output sensor suhu LM35, kondisi kedua yaitu 3 pompa dipicu oleh *output* sensor level yang apabila ketinggian air mencapai ketinggian yang diinginkan akan mencatu V_{cc} dengan V_{cc} yang telah

di pasang sesuai ketinggian level air yang diinginkan, dan kondisi terakhir yaitu 2 pompa dipicu oleh timer selama 60 detik, seperti yang dapat dilihat dalam Gambar 3.1.



Gambar 3.1 Smart Pump Plant

Keterangan :

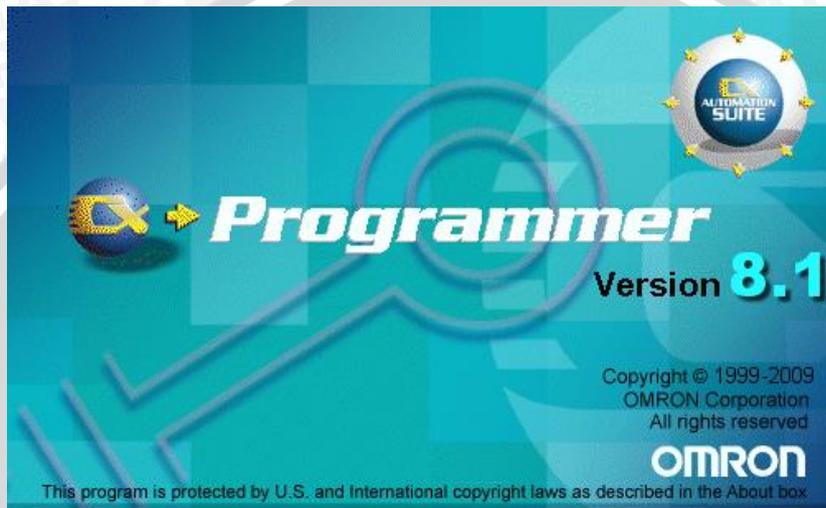
1. S0 : Saklar On – Off
2. A dan B : wadah air
3. C : Panel lampu indikator
4. M1 : Pompa 1 (S1)
5. M2 : Pompa 2 (S2)
6. M3 : Pompa 3 (S3)
7. M4 : Pompa 4 (LS1)
8. M5 : Pompa 5 (LS2)
9. M6 : Pompa 6 (LS3)
10. M7 : Pompa 7 (T1)
11. M8 : Pompa 8 (T2)
12. LS1 : Sensor ketinggian air 3 cm
13. LS2 : Sensor ketinggian air 6 cm
14. LS3 : Sensor ketinggian air 9 cm
15. S1 : Sensor suhu LM35
16. S2 : Sensor suhu LM35
17. S3 : Sensor suhu LM35

Semua pompa bekerja dengan tegangan yang sama, yaitu 24 volt. Tegangan 24 volt dibuat sama karena tegangan PLC adalah 24 volt, yang dimana PLC hanya akan bekerja dengan sensor dan aktuator yang memiliki tegangan input sebesar 24 volt.

3.2. Spesifikasi Alat

Spesifikasi alat yang digunakan pada skripsi ini adalah sebagai berikut :

1. Komputer yang sudah terinstall *software* CX – Programmer versi 8.1, seperti dalam Gambar 3.2.



Gambar 3.2 Program CX-One

2. *Power Supply Unit* 2560 (PSU) yang digunakan memiliki *range* 0-30 V. PSU berfungsi sebagai sumber tegangan catu daya PLC dimana tegangan *output* diatur sebesar 24 V, seperti dalam Gambar 3.3.



Gambar 3.3 Power Supply Unit 2560

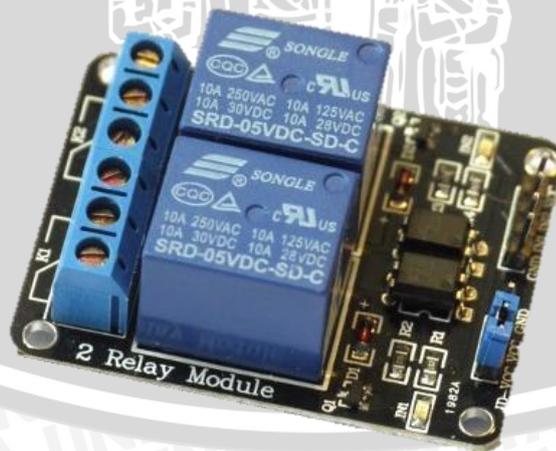
3. PLC yang digunakan adalah PLC OMRON tipe CP1L-L20DTI-D. PLC tipe CP1L adalah PLC yang umumnya digunakan di laboratorium karena dimensinya

yang cukup kecil dengan jumlah *port input/output* tidak terlalu banyak. PLC Omron tipe CP1L-LD20TI-D ditunjukkan dalam Gambar 3.4.



Gambar 3.4 PLC Omron CP1L-L20Dxx

4. Penggunaan *Relay* adalah karena output dari sensor *level* pada taraf tegangan 0 – 5 volt, sedangkan untuk PLC sendiri membutuhkan tegangan untuk konsisi *input* sebesar 24 volt. Oleh karena itu dibutuhkan *relay* untuk mengaktifkan atau memberikan sinyal ke PLC untuk memberikan logika *high* (24 volt) maupun *low* (0 volt), seperti yang ditunjukkan dalam Gambar 3.5.



Gambar 3.5 Relay 5 volt

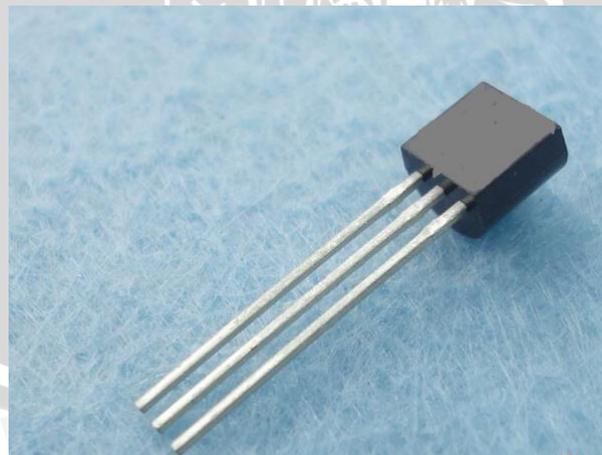
5. Pompa air ini digunakan sebagai aktuator pada sistem ini. Fungsi dari pompa air ini adalah untuk mengalirkan air ke dalam tangki. Pompa air yang digunakan

sebagai aktuator ini memiliki tegangan masukan sebesar 24V DC. Pompa air 24 V DC ditunjukkan dalam gambar 3.6.



Gambar 3.6 Pompa Air TIY – 152

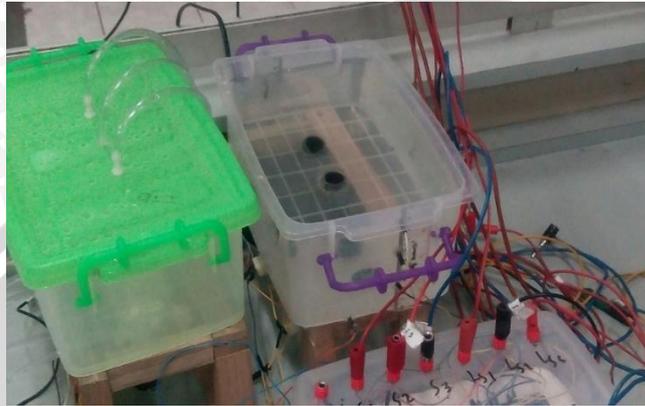
6. Sensor LM35 dengan *type* LM35 adalah sensor yang digunakan untuk mendeteksi suhu pada motor pompa air, seperti yang ditunjukkan dalam Gambar 3.7.



Gambar 3.7 Sensor LM35

7. *Smart pump plant* bekerja dengan catu daya 24 volt dan berfungsi sebagai *plant* pada sistem. *Plant* ini terdiri atas 8 pompa air yang dikombinasikan dengan 3

kondisi *input*, yaitu *input* dari sensor LM35, sensor *limit switch*, dan *timer*. *Smart pump plant* dapat dilihat dalam Gambar 3.8.



Gambar 3.8 *plant system*

3.3. Perancangan Kerja

Smart pump plant terdiri dari wadah plastik berbentuk persegi panjang, motor pompa DC 24v, sensor suhu, dan sensor *limit switch*. Untuk mengaktifkan alat secara keseluruhan, maka diperlukannya *input on-off* yang akan mengaktifkan dan mematikan keseluruhan sistem. Jika ingin mengaktifkan keseluruhan sistem, maka sakelar *on-off* haruslah berlogika 1 dan apabila saklar *on-off* berlogika 0 maka sistem keseluruhan akan mati.

Pertama - tama ketika alat aktif, maka motor pompa akan aktif. Lampu indikator akan menyala untuk memastikan pompa menyala. M1 akan menyala dan akan berhenti ketika suhu pompa mencapai $>50^{\circ}\text{C}$. Kemudian ketika M1 berhenti menyala, maka akan mengaktifkan M2. Seperti kondisi sebelumnya ketika suhu pada motor pompa mencapai $>50^{\circ}\text{C}$, maka pompa akan berhenti menyala. Kemudian ketika M2 berhenti menyala, maka akan mengaktifkan M3. M3 akan berhenti menyala jika suhu pada motor pompa mencapai $>50^{\circ}\text{C}$ dan kemudian akan kembali mengaktifkan M1 ketika M3 berhenti menyala. Keadaan ini terus berulang sesuai teori kontrol sekuensial (bergantian). S1,S2,dan S3 yang tadinya berlogika 1 jika suhu $>50^{\circ}\text{C}$ dan akan kembali berlogika 0 jika suhu $<50^{\circ}\text{C}$.

M4 akan menyala jika LS1 aktif, LS1 akan aktif atau berlogika 1 jika air naik melewati sensor $\geq 3\text{cm}$ dari dasar wadah dan menggabungkan V_{cc} dengan V_{cc} pada sensor. LS1 akan mati jika ketinggian air $< 3\text{cm}$. Jika LS1 mati maka M4 akan berhenti menyala.

M5 akan menyala jika LS2 aktif. LS2 aktif atau berlogika 1 jika air naik melewati sensor ≥ 6 cm dari dasar wadah dan menggabungkan V_{cc} dengan V_{cc} pada sensor. LS2 akan mati jika ketinggian air < 6 cm. Jika LS2 mati maka M5 akan berhenti menyala. M6 akan menyala jika LS3 aktif. LS3 aktif atau berlogika 1 jika air naik melewati sensor ≥ 9 cm dari dasar wadah dan menggabungkan V_{cc} dengan V_{cc} pada sensor. LS3 akan mati jika ketinggian air < 9 cm. Jika LS3 mati maka M6 akan berhenti menyala. Siklus ini juga terus berkelanjutan seperti teori sistem kontrol sekuensial

Kemudian selain LS dan S, yang terakhir adalah T. Timer digunakan untuk menentukan lamanya waktu menyala pada motor pompa. Setelah alat aktif, maka akan mengaktifkan M7, kemudian timer menghitung selama 60 detik dan kemudian memberhentikan M7. Kemudian ketika M7 berhenti menyala, maka akan mengaktifkan M8, sama halnya seperti logika sebelumnya timer menghitung selama 60 detik dan akan memberhentikan M8 dan setelah timer aktif selama 60 detik, M7 akan aktif kembali. Jika M8 berhenti menyala, maka akan mengaktifkan M7. Terus menerus berulang hingga alat di berhentikan (S0 berlogika 0).

3.4. Tabel Alamat *Input* dan *Output*

Setelah didapatkan deskripsi operasi otomatisasi yang diinginkan, maka selanjutnya adalah mendata langkah, transisi, dan intruksi pendukung, serta memberi nama dan simbol secara jelas. Dalam pembuatan bagan grafcet dibutuhkan data *input* dan data *output* dari PLC, yang data tersebut akan membantu mempermudah pembuatan dari bagan grafcet. Alamat *output* pada PLC dapat dilihat dalam Tabel 3.1.

Tabel 3.1 Alamat *Output* Perangkat Lunak Grafcet

No.	Alamat	Keterangan	Simbol
1	Q 100.00	Motor Pompa Suhu 1	M1
2	Q 101.00	Motor Pompa Suhu 2	M2
3	Q 102.00	Motor Pompa Suhu 3	M3
4	Q 103.00	Motor Pompa <i>Limit Switch</i> 1	M4
5	Q 104.00	Motor Pompa <i>Limit Switch</i> 2	M5
6	Q 105.00	Motor Pompa <i>Limit Switch</i> 3	M6
7	Q 106.00	Motor Pompa Timer 1	M7
8	Q 107.00	Motor Pompa Timer 2	M8
9	Q 110.00	Lampu Indikator 1	LI1
10	Q 111.00	Lampu Indikator 2	LI2
11	Q 112.00	Lampu Indikator 3	LI3

12	Q 113.00	Lampu Indikator 4	LI4
13	Q 114.00	Lampu Indikator 5	LI5
14	Q 115.00	Lampu Indikator 6	LI6
15	Q 116.00	Lampu Indikator 7	LI7
16	Q 117.00	Lampu Indikator 8	LI8

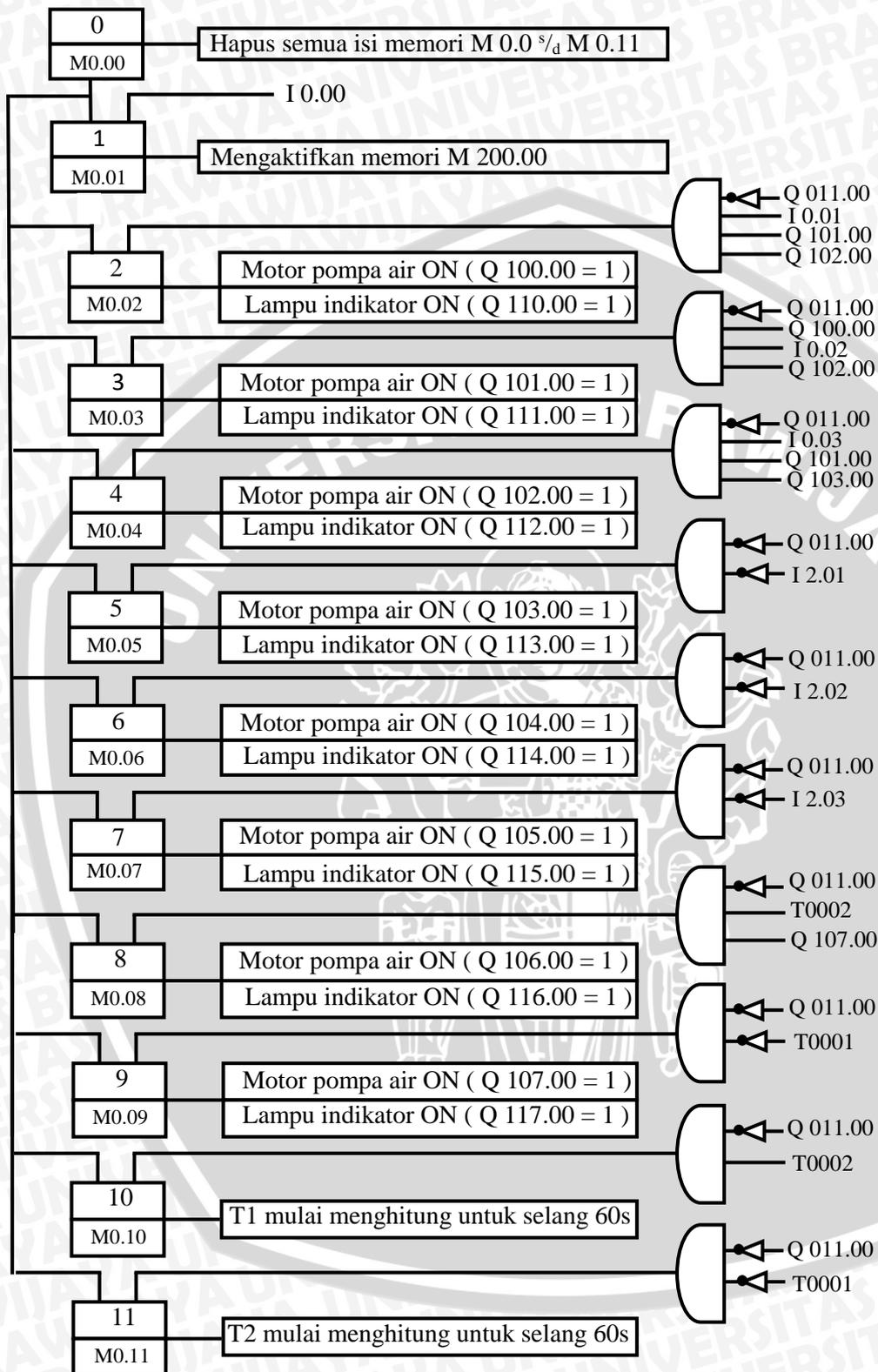
Sedangkan data *input* yang dibutuhkan untuk mempermudah dalam pembuatan bagan grafcet adalah data dari alamat *input* pada PLC yang dapat dilihat dalam Tabel 3.2.

Tabel 3.2 Alamat *Input* Perangkat Lunak Grafcet

No.	Alamat	Keterangan	Simbol
1	I 0.00	Saklat On Off	S0
2	I 0.01	Sensor Suhu 1	S1
3	I 0.02	Sensor Suhu 2	S2
4	I 0.03	Sensor Suhu 3	S3
5	I 2.01	Sensor Limit Switch 1	LS1
6	I 2.02	Sensor Limit Switch 1	LS2
7	I 2.03	Sensor Limit Switch 1	LS3
8	T0001	Sensor Timer 1	T1
9	T0002	Sensor Timer 2	T2

3.5 Perancangan Diagram State

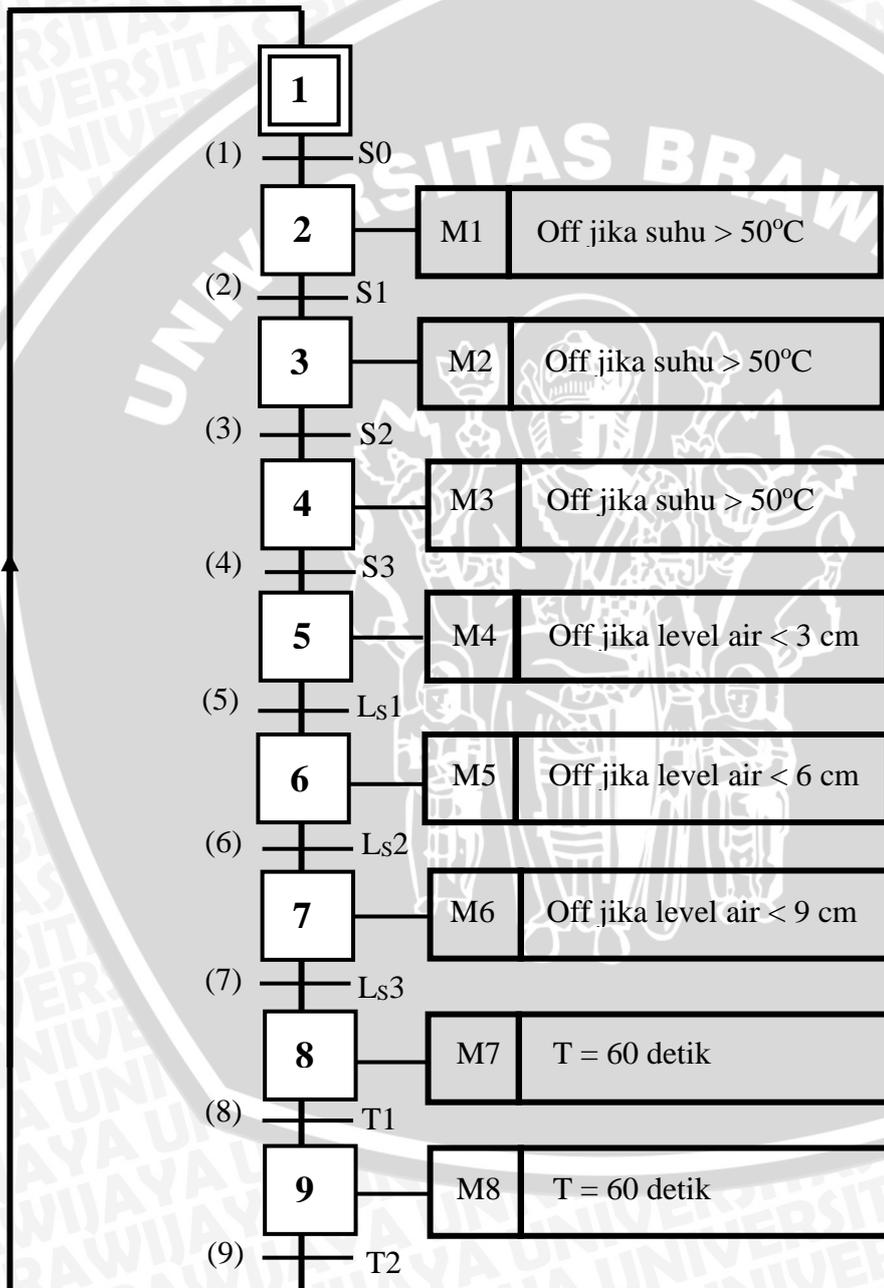
Diagram state digunakan untuk mempermudah dalam memahami sistem kerja alat, dengan didasari alamat *input* dan *output* maupun sistem kerja alat, telah dihasilkan sebuah *diagram state* yang nantinya dijadikan sebuah acuan dalam pembuatan bagan grafcet dan *ladder diagram*. *Diagram state* alat dapat dilihat dalam Gambar 3.9.



Gambar 3.9 Perancangan diagram state

3.6 Perancangan Metode Grafcet

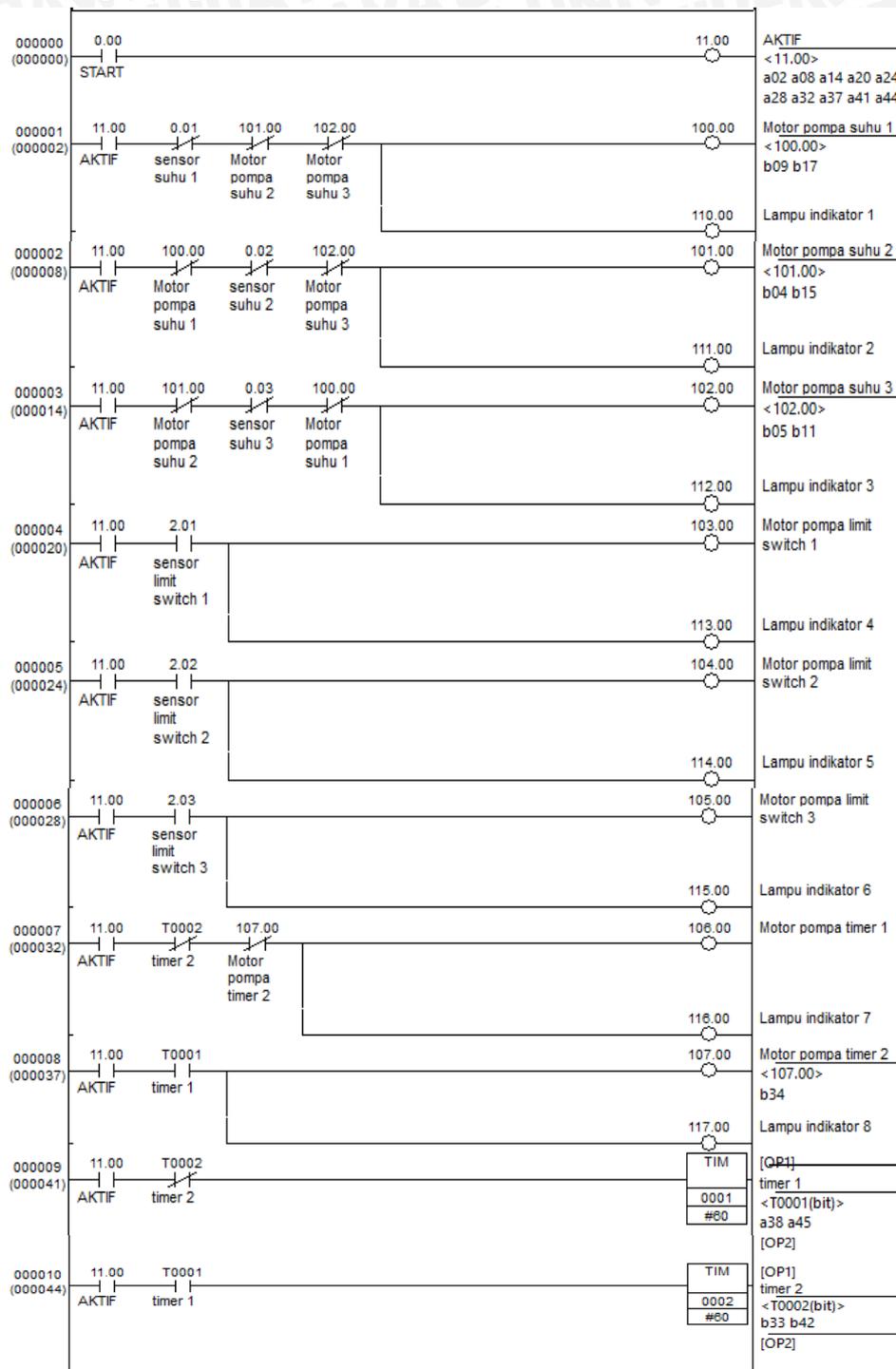
Metode grafcet dibuat untuk mempermudah dalam perancangan *ladder diagram* pada PLC. Pembuatan bagan grafcet ini didasari oleh data *input* dan *output* dari PLC. Bagan ini dibuat agar dapat dengan mudah mengetahui dan mengerti alur kerja dari alat, karena PLC di program dengan CX-Programmer yang dimana *software* ini menggunakan *ladder* untuk memprogram dan mengubah logika di dalam sebuah PLC. Hasil perancangan grafcet dapat dilihat dalam Gambar 3.10.



Gambar 3.10 Perancangan metode grafcet

3.7 Perancangan Ladder Diagram

Untuk kebutuhan perancang desain rangkaian listrik, penggunaan PLC dengan *ladder diagram* sangat membantu mengurangi kesulitan dalam pengoperasian rangkaian listrik dalam panel kontrol, sehingga menjadi ringkas, kompleks, dan sangat fleksibel, seperti yang ditampilkan dalam Gambar 3.11.

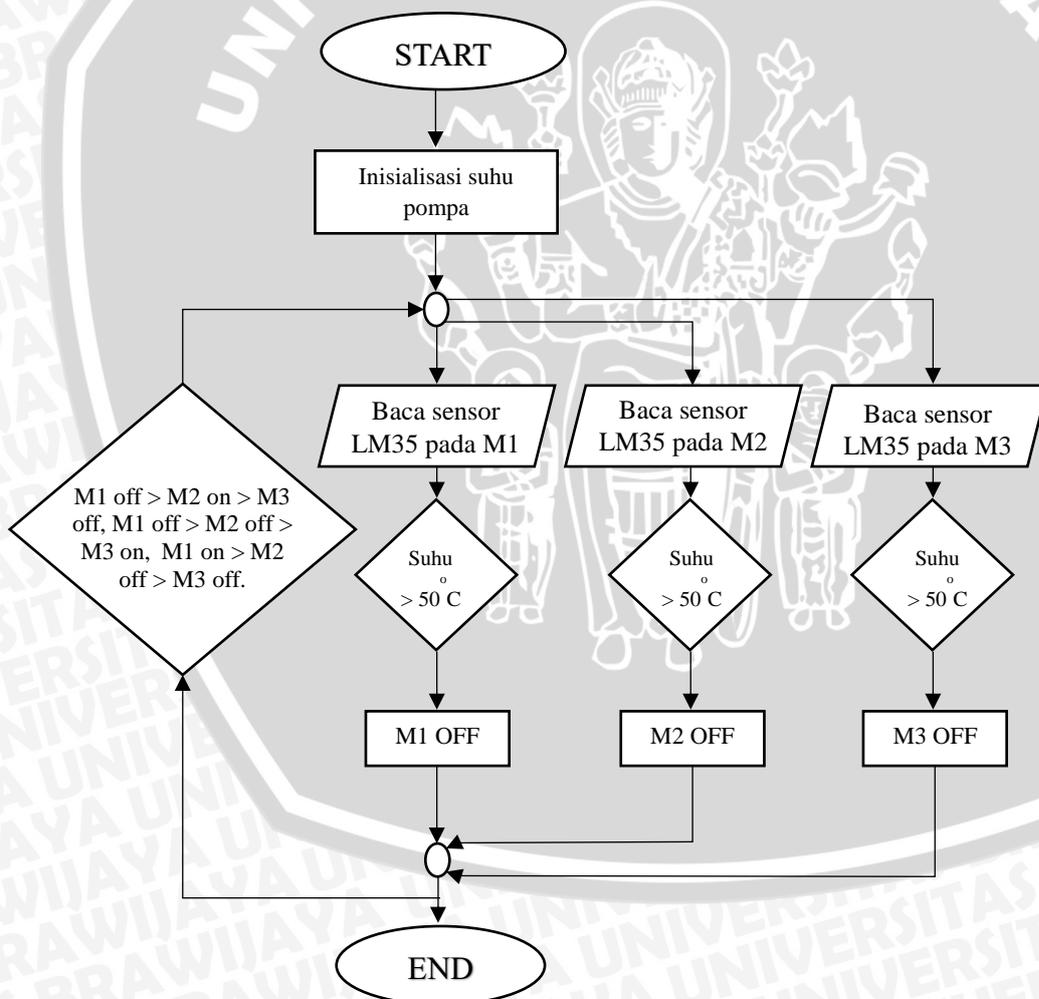


Gambar 3.11 Ladder diagram

Jika melakukan modifikasi ataupun *upgrade* sistem dari rangkaian listrik mesin yang dirancang tidak perlu kesulitan, karena *ladder diagram* dapat langsung diganti melalui komputer dengan memasukkan *input* logika kedalam PLC melalui perantara komputer

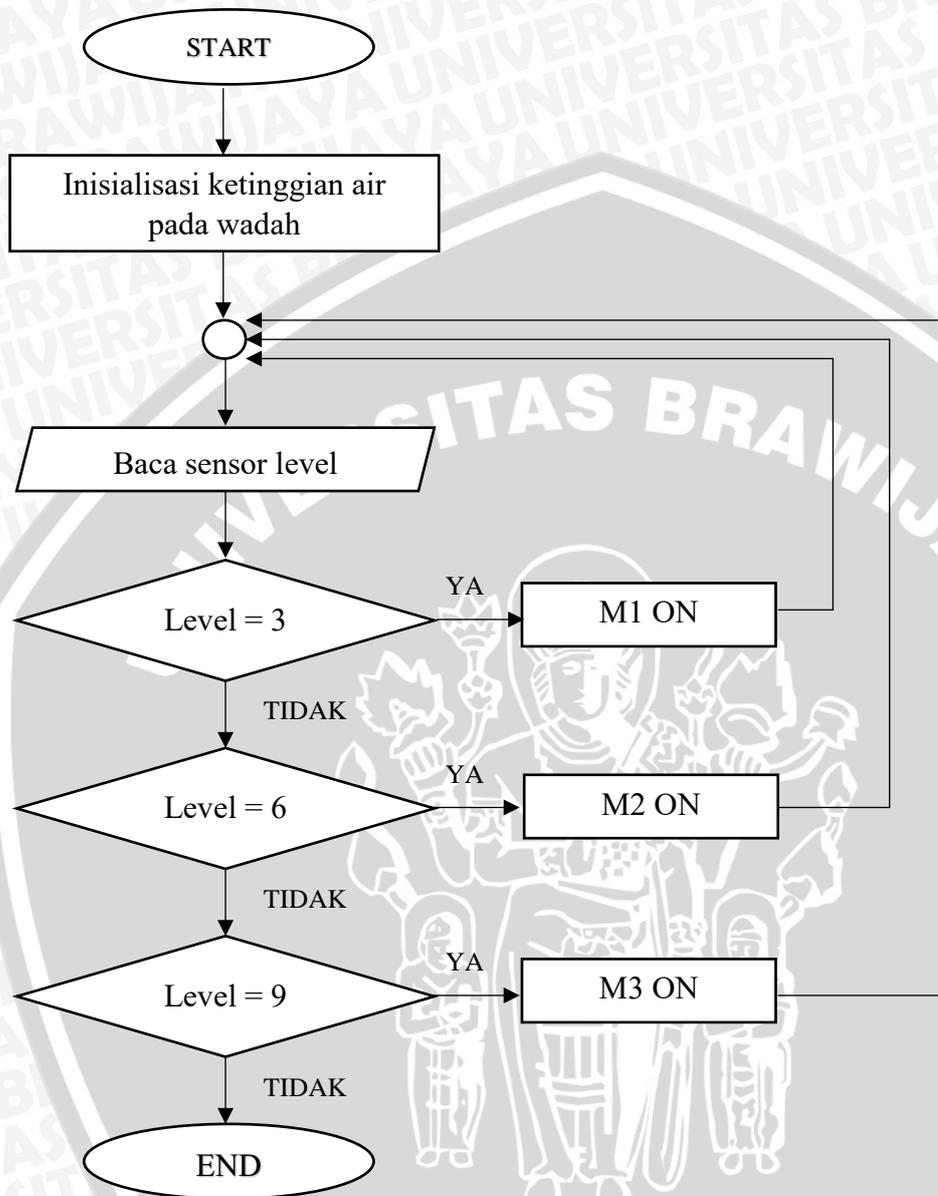
3.8 Diagram Alir Sensor

Setelah semua komponen pada alat sudah terhubung sesuai dengan alur *diagram state*, maupun sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem telah dibuat. Maka dari itu dibuatlah suatu diagram alir program yang merupakan gambaran alur proses program yang dilaksanakan oleh kontroler pada saat implementasi, dengan kontroler yang digunakan adalah PLC. Pada diagram alir penelitian ini akan dibagi 3 yaitu, pertama diagram alir sensor LM35. Berikut ini adalah diagram alir sensor LM35 yang ditampilkan dalam Gambar 3.12.



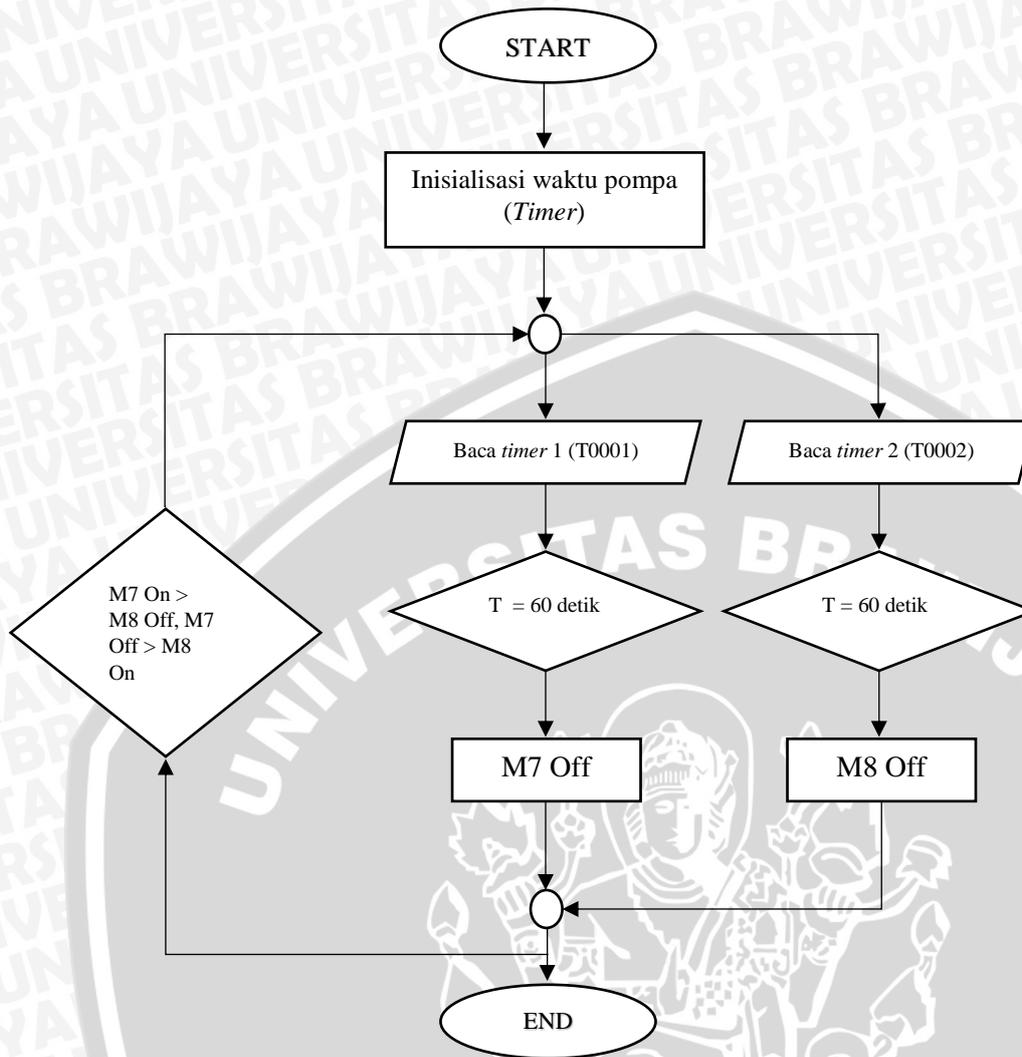
Gambar 3.12 Diagram alir sensor suhu.

Kedua yaitu diagram alir sensor *limit switch*. Berikut ini adalah diagram alir sensor *limit switch* yang ditampilkan dalam Gambar 3.13.



Gambar 3.13 Diagram alir sensor level.

Ketiga yaitu diagram alir *timer*. Berikut ini adalah diagram alir sensor *timer* yang ditampilkan dalam Gambar 3.14.



Gambar 3.14 Diagram alir timer.

3.9 Pengambilan kesimpulan

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapat telah sesuai dengan yang direncanakan sebelumnya, maka sistem kontrol sekuensial tersebut telah berhasil dan memenuhi harapan.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas mengenai hasil pengujian sistem dan pembahasan hasil pengujian untuk mengetahui keberhasilan alat yang dibuat. Pengujian Alat Berdasarkan Ladder Diagram yang dibuat dari Metode Grafcet.

Pengujian perangkat lunak ini bertujuan untuk mengetahui apakah perangkat lunak yang dibuat telah dapat bekerja sesuai dengan metode Grafcet yang telah direncanakan.

Langkah – langkah pengujian perangkat lunak adalah sebagai berikut :

1. Menghubungkan terminal *input* PLC dengan *input* eksternal.
2. Menghubungkan terminal *output* PLC dengan *output* eksternal.
3. Menulis perangkat lunak dalam bentuk ladder diagram berdasarkan metode Grafcet menggunakan CX – One.
4. Mengisi PLC dengan perangkat lunak yang telah dibuat melalui kabel komunikasi.
5. Mengoperasikan sistem motor pompa (*Smart Pump Plant*).

Setelah semua langkah dan proses dilakukan, maka akan didapatkan sebuah hasil data bahwa alat bekerja dengan sesuai rencana atau tidak sesuai rencana. Didapatkan hasil sesuai data *ladder diagram* adalah sama seperti apa yang dihasilkan pada alat, seperti yang ditampilkan dalam Gambar 4.1.



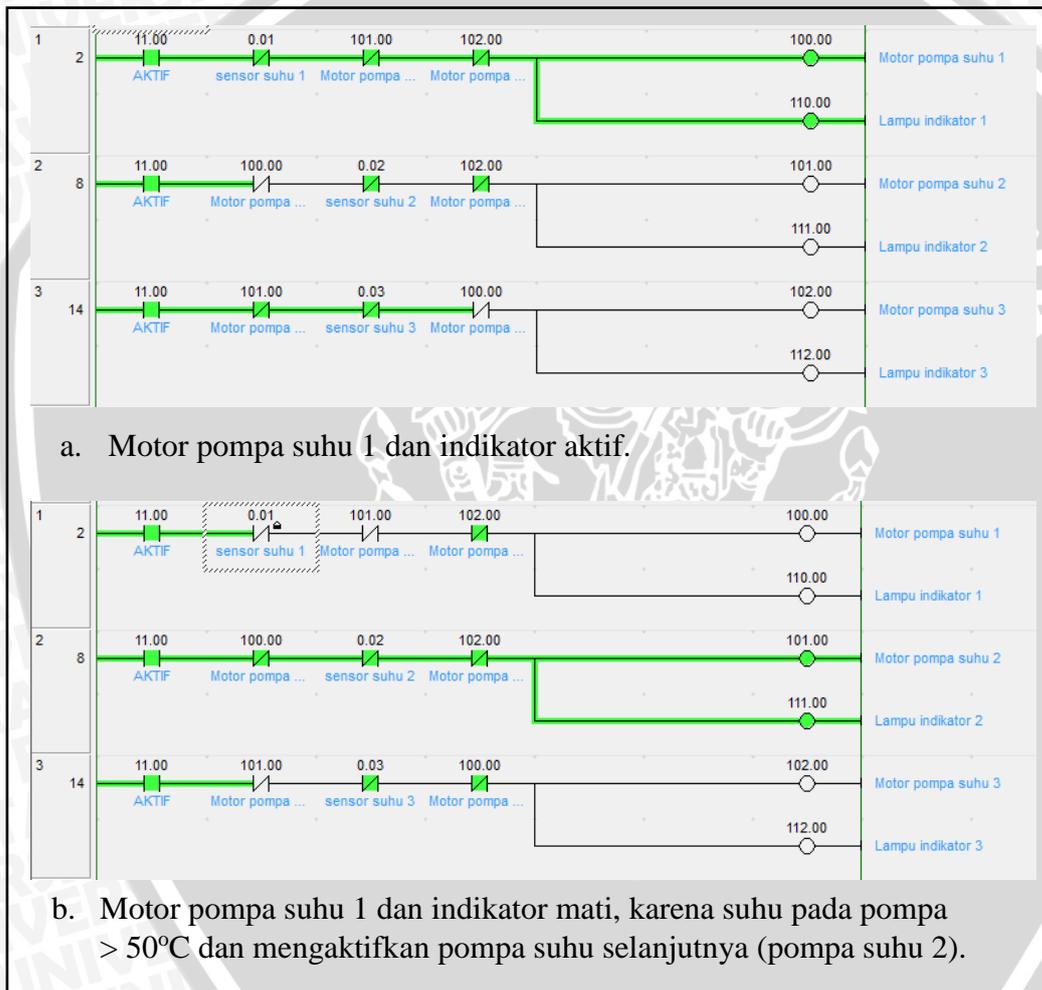
Gambar 4.1 Sakelar pada tampilan CX - Programmer

Berikut ini adalah hasil dari pengujian yang dilakukan pada smart pump plant yang terhubung langsung dengan komputer, seperti yang ditampilkan dalam Tabel 4.1.

Tabel 4.1 Hasil Pengujian Saklar Pada *Smart Pump Plant* Dengan Metode Grafcet.

No.	Input PLC	Output PLC
1	000.00 (Tombol tekan <i>start</i> = 1)	011.00 <i>ON</i> (Alat aktif)
2	000.00 (Tombol tekan <i>start</i> = 0)	011.00 <i>OFF</i> (Alat tidak aktif)

Setelah sakelar diaktifkan maka pompa dan indikator akan aktif, tetapi jika suhu pompa > 50°C maka pompa dan lampu indikator akan mati, seperti yang ditampilkan dalam Gambar 4.2.

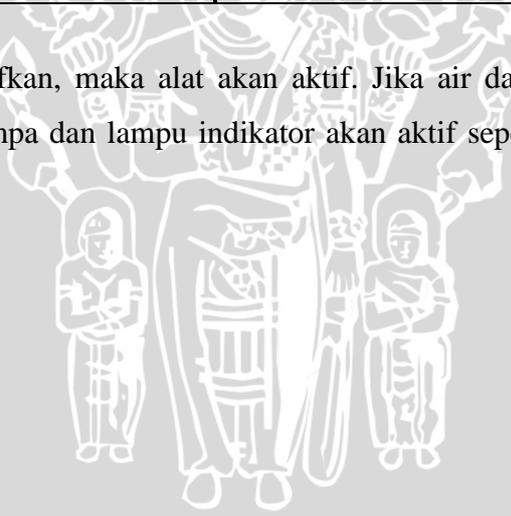
**Gambar 4.2** Logika suhu pada tampilan CX - Programmer

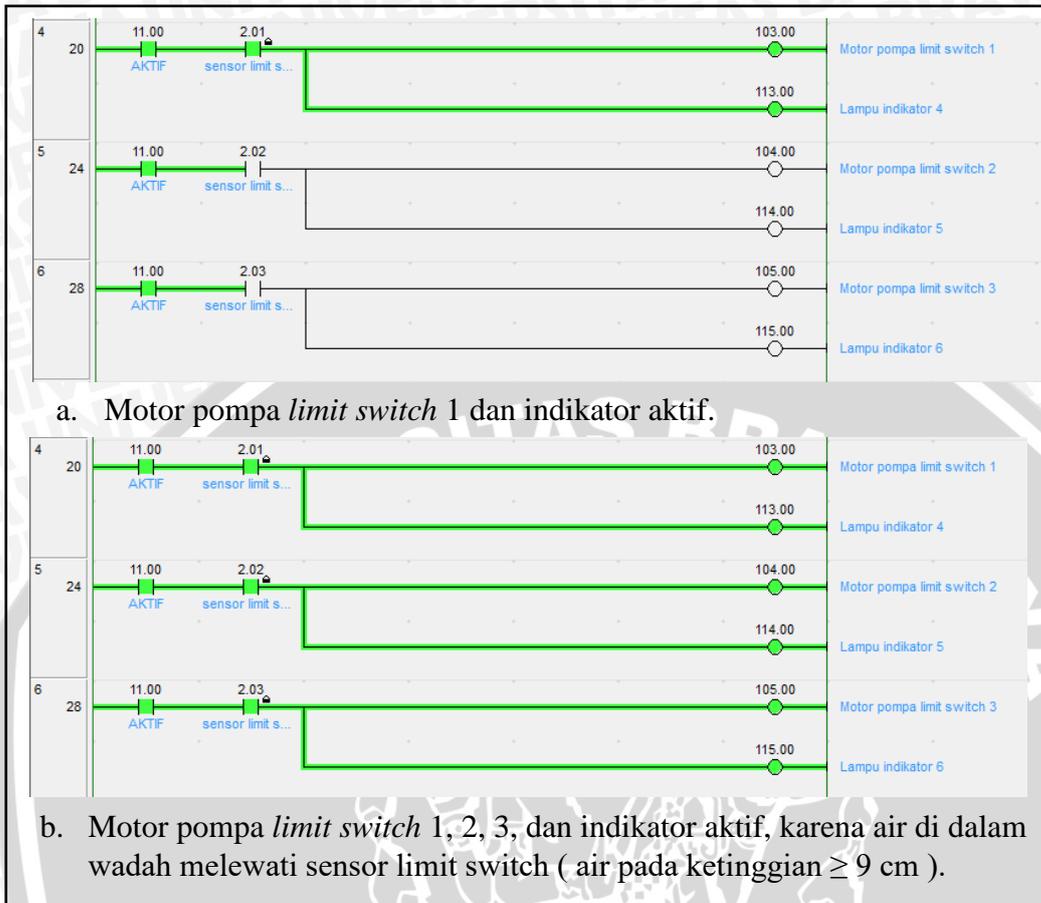
Berikut ini adalah hasil dari pengujian yang dilakukan pada smart pump plant yang terhubung langsung dengan komputer, seperti yang ditampilkan dalam Tabel 4.2.

Tabel 4.2 Hasil Pengujian Sensor Suhu Pada *Smart Pump Plant* Dengan Metode Grafcet.

No.	Input PLC	Output PLC
1	000.00 (Tombol tekan <i>start</i> = 1)	011.00 ON(Alat aktif)
2	011.00 ON (berlogika 1)	100.00 dan 110.00 ON (Pompa dan Led Aktif).
3	000.01 ON (berlogika 1)	100.00 dan 110.00 OFF
4	100.00 ON (berlogika 0)	101.00 dan 111.00 ON (Pompa dan Led Aktif).
5	000.02 ON (berlogika 1)	101.00 dan 111.00 OFF
6	101.00 ON (berlogika 0)	102.00 dan 112.00 ON (Pompa dan Led Aktif).
7	000.03 ON (berlogika 1)	102.00 dan 112.00 OFF
8	102.00 ON (berlogika 0)	100.00 dan 110.00 ON (Pompa dan Led Aktif).

Setelah sakelar diaktifkan, maka alat akan aktif. Jika air dalam wadah melewati sensor *limit switch* maka pompa dan lampu indikator akan aktif seperti yang ditampilkan dalam Gambar 4.3





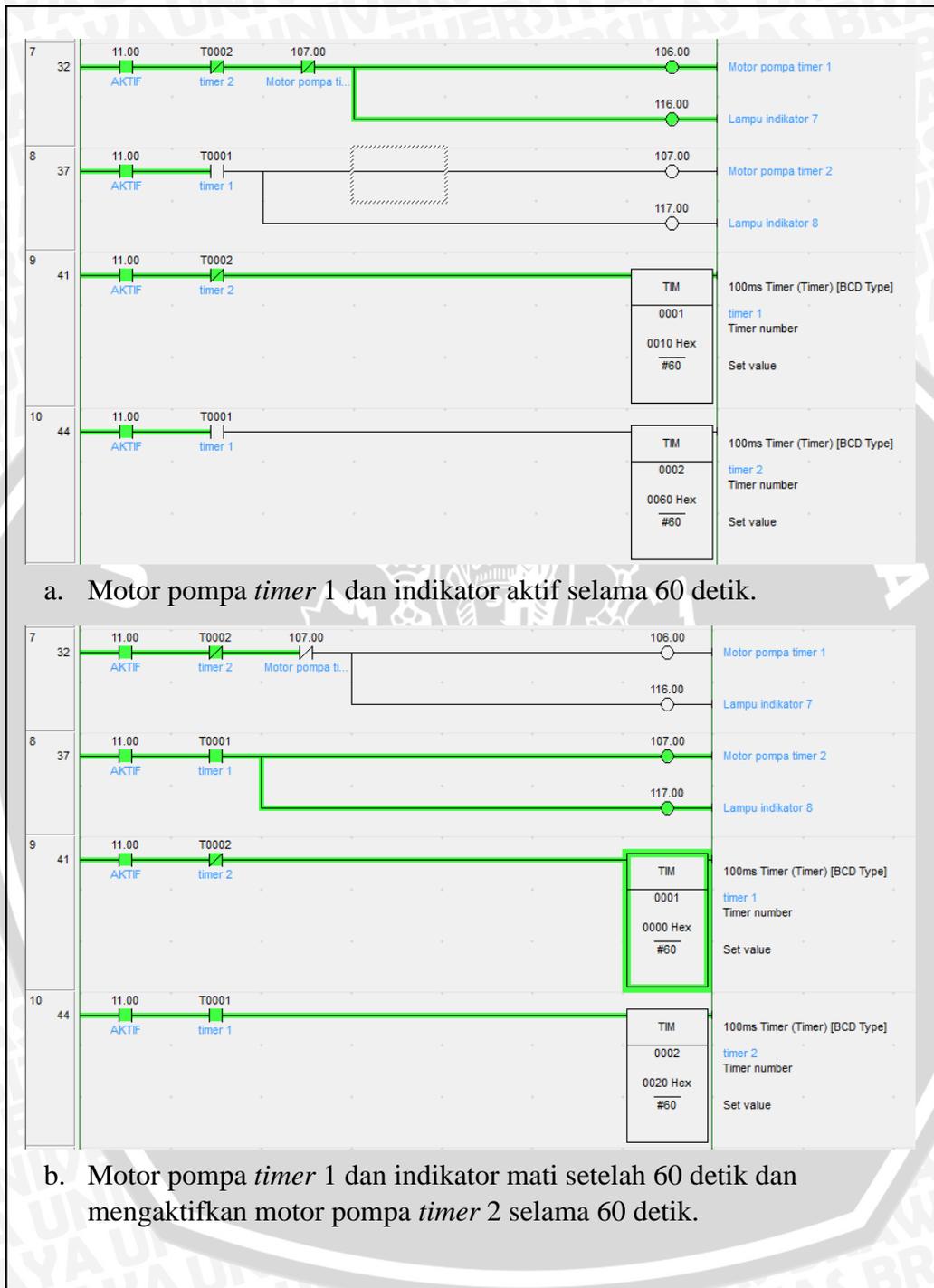
Gambar 4.3 Logika *limit switch* pada tampilan CX - Programmer

Berikut ini adalah hasil dari pengujian yang dilakukan pada smart pump plant yang terhubung langsung dengan komputer, seperti yang ditampilkan dalam Tabel 4.3.

Tabel 4.3 Hasil Pengujian Sensor Level Air Pada *Smart Pump Plant* Dengan Metode Grafcet.

No.	Input PLC	Output PLC
1	000.00 (Tombol tekan <i>start</i> = 1)	011.00 ON (Alat aktif)
2	002.01 ON (berlogika 1)	103.00 ON (Pompa aktif)
3	002.02 ON (berlogika 1)	104.00 ON (Pompa aktif)
4	002.03 ON (berlogika 1)	105.00 ON (Pompa aktif)

Setelah sakelar diaktifkan, maka akan mengaktifkan pompa *timer* 1 dan indikator aktif, kemudian akan mati setelah 60 detik dan mengaktifkan pompa *timer* 2 selama 60 detik seperti yang ditampilkan dalam Gambar 4.4.



Gambar 4.4 Logika *timer* pada tampilan CX - Programmer

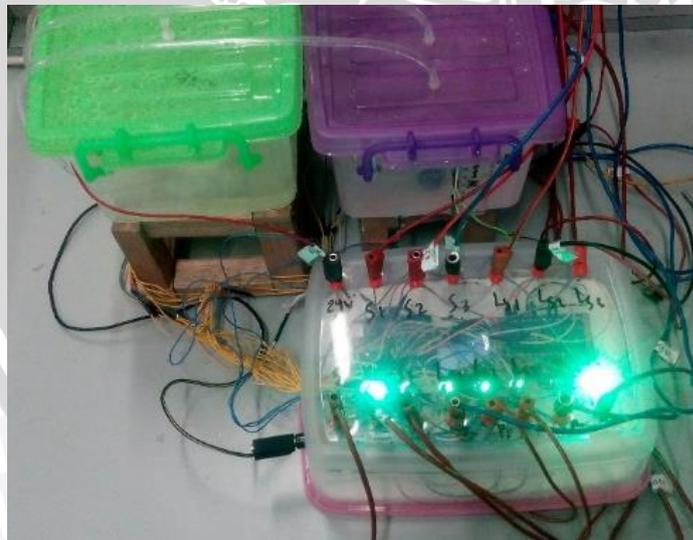
Berikut ini adalah hasil dari pengujian yang dilakukan pada smart pump plant yang terhubung langsung dengan komputer, seperti yang ditampilkan dalam Tabel 4.4.



Tabel 4.4 Hasil Pengujian Timer Pada *Smart Pump Plant* Dengan Grafcet.

No.	Input PLC	Output PLC
1	000.00 (Tombol tekan <i>start</i> = 1)	011.00 ON(Alat aktif)
2	011.00 ON (berlogika 1)	106.00 ON, 116.00 ON, dan TIM 1 ON (60 detik)
3	T0001 (berlogika 1) → setelah 60 detik	107.00 ON, 117.00 ON, 106.00 OFF, 116.00 OFF, danTIM 2 ON (60 detik)
4	T0002 (berlogika 1) → setelah 60 detik	106.00 ON, 116.00 ON, 107.00 OFF, 117.00 OFF, danTIM 1 ON (60 detik)

Metode grafcet yang dihasilkan mempunyai bentuk yang sederhana dan mudah dimengerti. Grafcet menghasilkan sistem yang berurutan (*sequence*), sehingga pengoprasian sistem harus sesuai urutan, jika sistem dioprasikan secara tidak berurutan maka sistem tidak mengalami perubahan keadaan dan sensor tidak bekerja sesuai yang diharapkan, seperti yang ditampilkan dalam Gambar 4.5.

**Gambar 4.5** *Smart pump plant* telah bekerja

Sistem pompa air telah bekerja sesuai dengan yang direncanakan, seperti yang tergambar jelas dalam metode Grafcet yang di aplikasikan pada *ladder dia*

BAB V

PENUTUP

5.1. KESIMPULAN

Dari perancangan, pembuatan dan pengujian sistem otomasi pada alat *smart pump plant* dengan menggunakan *Programmable Logic Controller* (PLC) dapat diambil kesimpulan sebagai berikut :

1. Perancangan menggunakan metode *grafcet* sangatlah membantu di dalam memudahkan pembacaan sistem PLC, karena *grafcet* memiliki bentuk yang sederhana dan mudah dimengerti. Karena *grafcet* menghasilkan sistem yang berurutan, maka metode *grafcet* sangat dibutuhkan dalam menghasilkan hasil yang optimal.
2. Metode *grafcet* dan kontrol sekuensial telah membuat alat *smart pump plant* sesuai dengan yang diharapkan yaitu suhu akan mematikan pompa jika suhu > 50°C, level yang akan menyalakan pompa jika sensor *limit switch* terkena air dengan tinggi 3 cm, 6cm, 9 cm, dan yang selanjutnya adalah kondisi *timer* yang akan menyalakan pompa selama 60 detik.

5.2. SARAN

Karena dalam pembuatan sistem ini masih dirasa jauh dari kesempurnaan yaitu tidak dapat di kontrol dan di monitoring secara langsung, maka sangat dimungkinkan untuk dilakukan pengembangan dan penyempurnaan. pengembangan dan penyempurnaan yang dimaksudkan yaitu, pengembangan alat *smart pump plant* menggunakan *Human Machine Interface* (HMI). Karena dengan HMI dapat memonitoring kinerja pompa agar lebih maksimal.



DAFTAR PUSTAKA

- Ardiansyah, H. 2013. Perancangan Simulator Sistem Pengepakan Dan Penyortiran Barang Berbasis PLC. *JurnalRekaElkomika*. I (4):373.
- Arifin, M. 2016. Efektivitas Sistem Pengaman Pompa Air Berbasis PLC. *Skripsi*. Dipublikasikan. Semarang: Universitas Negeri Semarang.
- Budiyanto, M. 2011. *Wijaya, Pengenalan Dasar-dasar PLC(Programmable Logic Controller)*. Yogyakarta: Gava Media.
- Effendi, A. 2013. Perancangan Pengontrolan Pemanas Air Menggunakan PLC. *JurnalTeknikElektro*. II (3).
- Faisal, M. 2016. Perancangan Dan Pengujian Alat Uji Pompa Seri Dan Paralel. *Skripsi*. Dipublikasikan. Bandar Lampung: Universitas Lampung.
- Haase, K. 2010. *Programmable Logic Controls PLC III*. German: Leybold Didactic GMBH.
- Irjan, L. 2013. Sistem Pengendali Cairan Tinta Printer Epson C90 Sebagai Simulasi Pada Mesin Percetakan Berbasis PLC. *JurnalPenelitian*. X (2):158-163.
- Prayogo, R. 2012. Pengaturan PWM (*Pulse Width Modulation*) Dengan PLC. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Rusli, M. 2012. *Pengantar Analisis dan Desain PLC*. Malang: UBPress.
- Syaprudin. 2010. Sistem Kendali Sekuensial Perajang Ketela Pohon. *JurnalPoliteknologi*. IX (2):158-164.
- Tri, A. 2011. Implementasi Mikrokontroler Sebagai Pengukur Suhu Delapan Ruangan. *JurnalTeknologi*. IV (2):153-159.
- Wicaksono, H. 2009. *Teori, Pemograman dan aplikasi dalam otomasi sistem*. Yogyakarta: Graha Ilmu.
- Yusran, A. 2013. *Elektronika dasar : Limit switch dan saklar tekan*. Jakarta: Bumi Aksara.

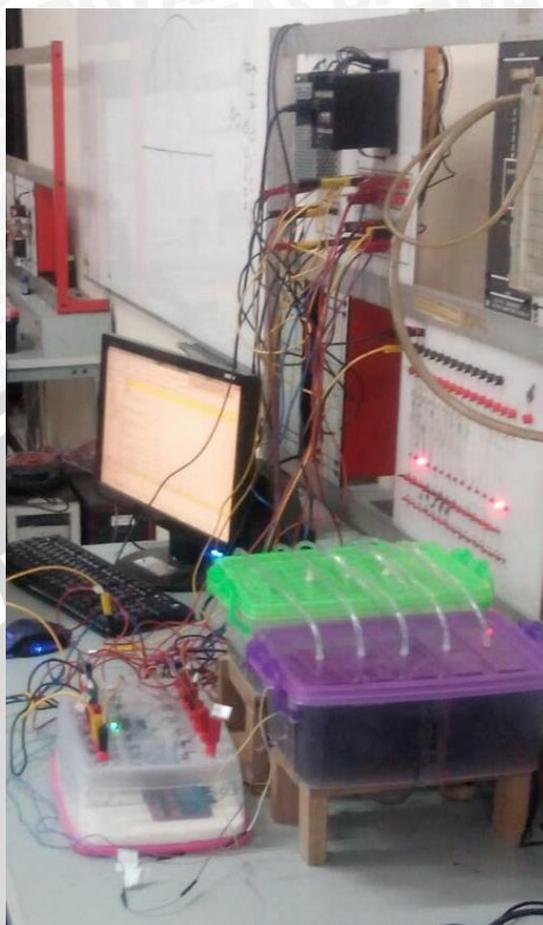


UNIVERSITAS BRAWIJAYA



LAMPIRAN

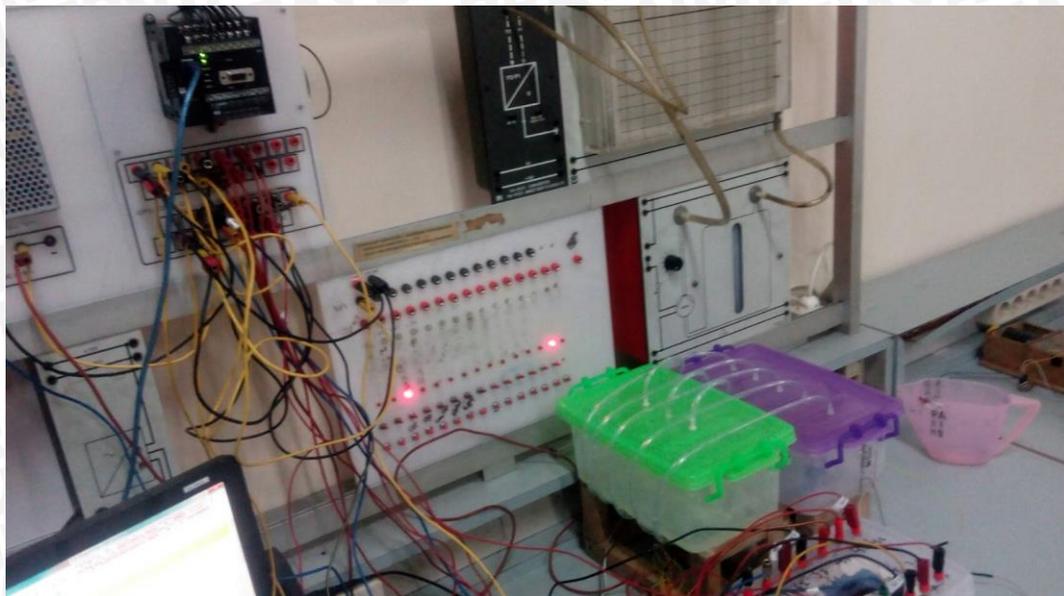
Foto Alat



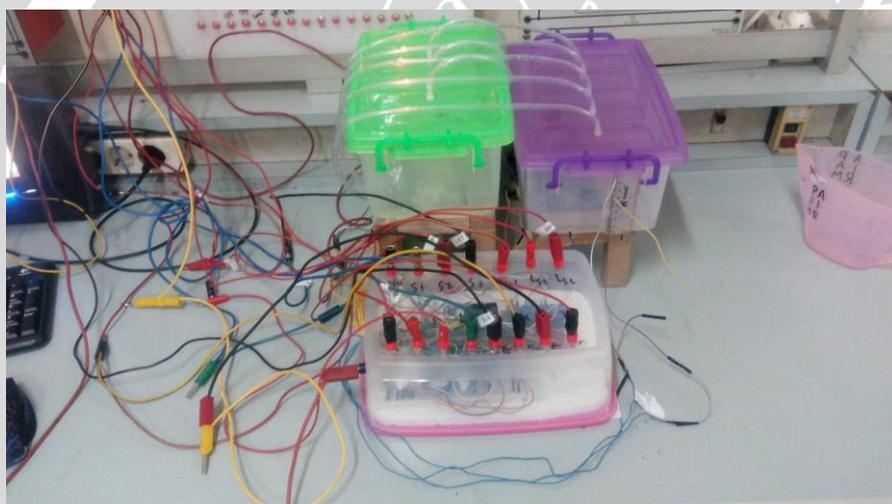
Alat tampak samping



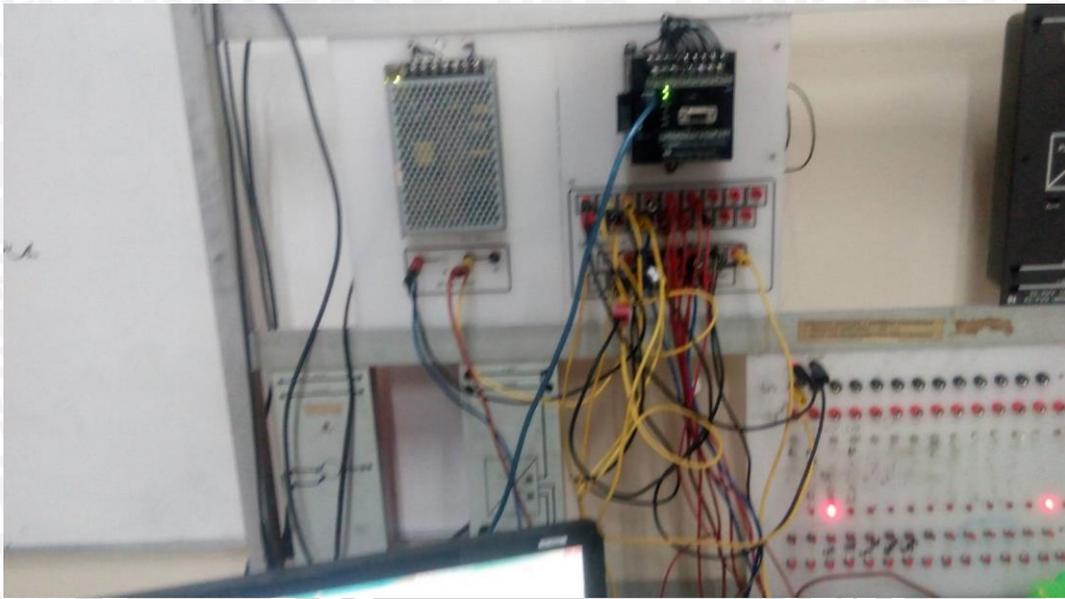
Alat tampak depan



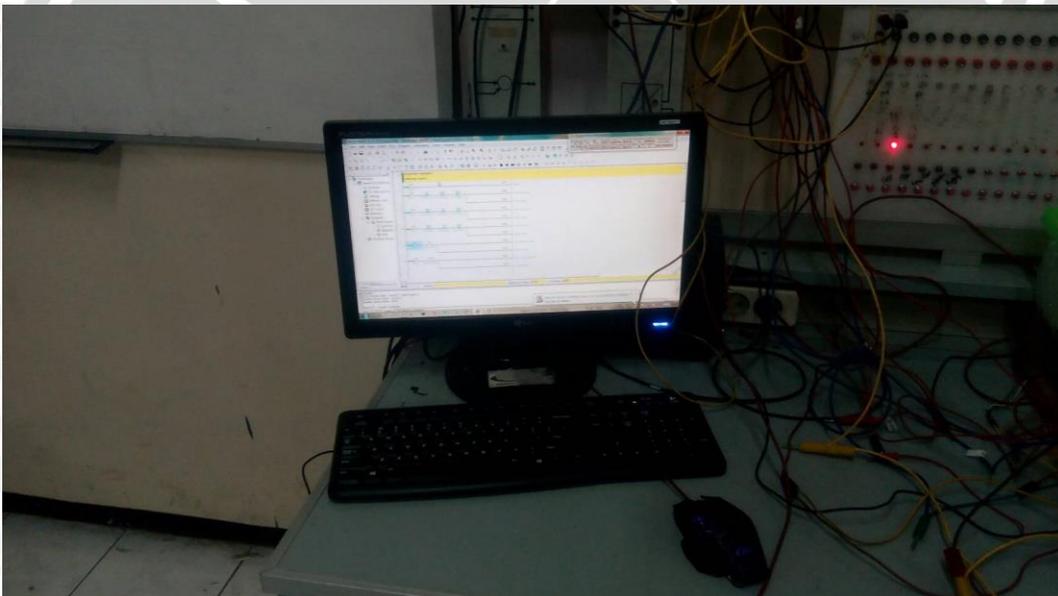
Alat tampak atas



Smart pump plant telah terhubung dengan PLC



Alat dan PC telah terhubung dengan PLC



PLC telah terhubung dengan software CX-Programmer

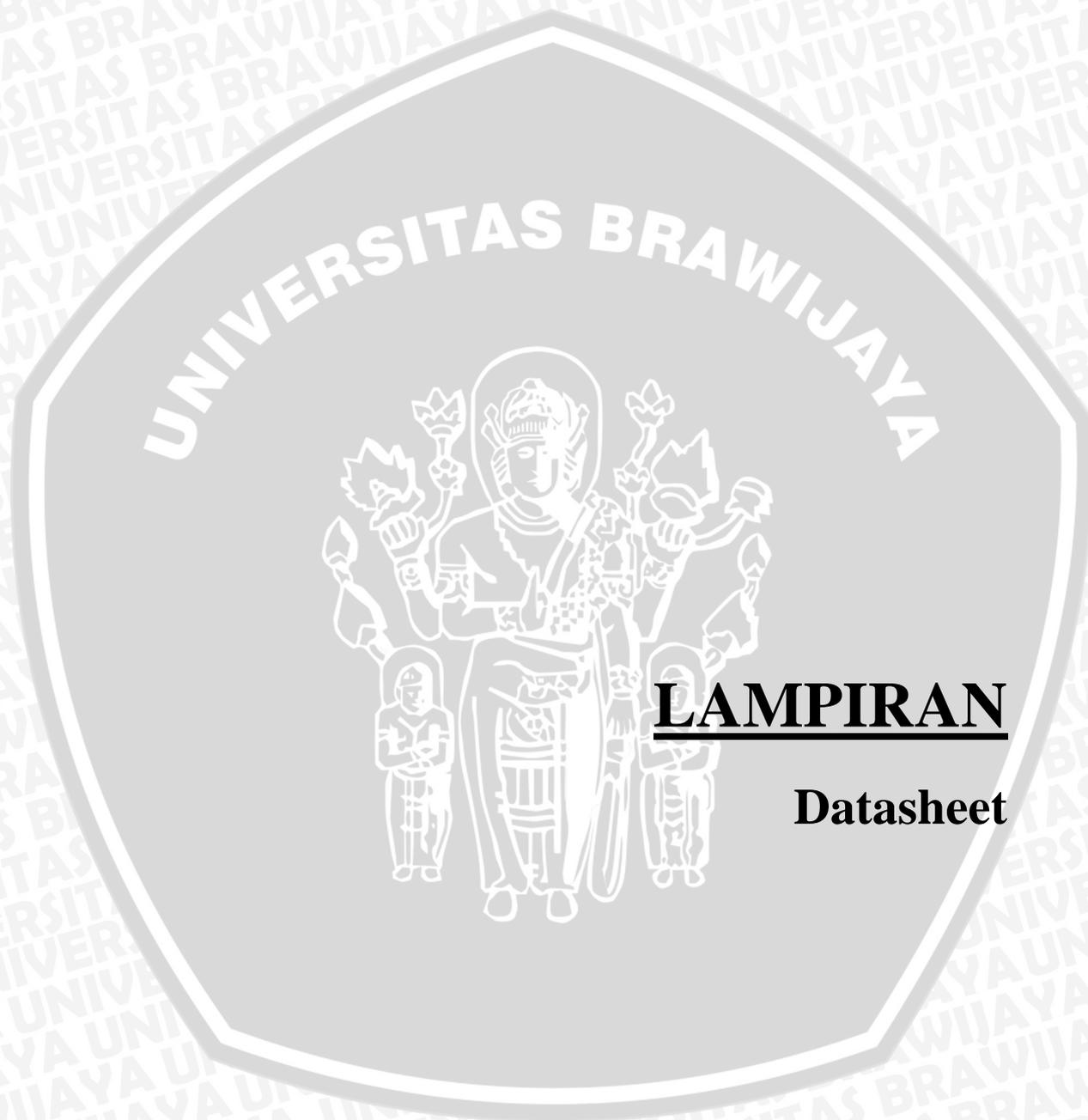
UNIVERSITAS BRAWIJAYA



LAMPIRAN

Data Pompa





LAMPIRAN

Datasheet