BAB III

METODE PENELITIAN

3.1 Perancangan

Perancangan pada bab III terdiri dari perancangan perangkat keras dan perancangan perangkat lunak, dengan adanya perancangan perangkat keras dan perangkat lunak akan mempermudah dalam pembuatan sistem yang akan dibuat.

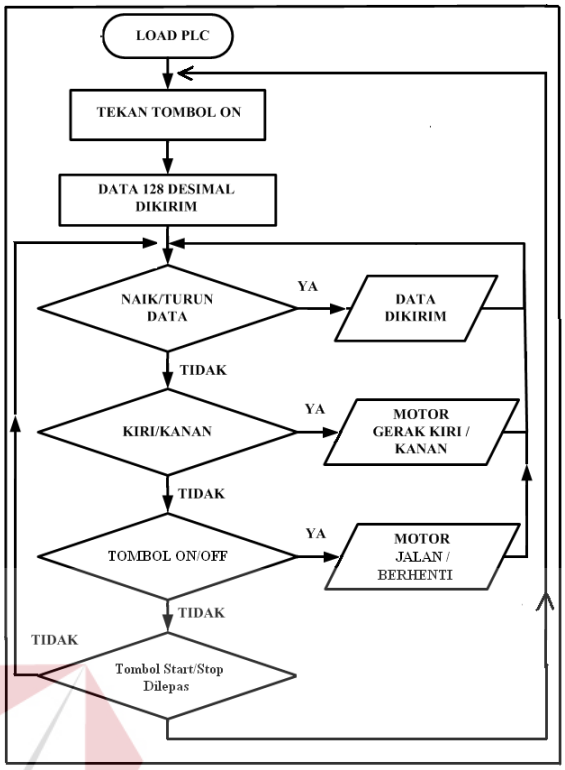
3.1.1 Perancangan Perangkat Keras

Pada perancangan dan pembuatan perangkat keras ini, langkah pertama yang harus ditentukan adalah menentukan tujuan dari pembuatan sistem ini. Kemudian merancang blok diagram secara umum atau skematik rangkaian.

Perangkat utama pada alat yang didisain oleh penulis adalah PLC dan Modul konverter, PLC sebagai pengontrol serta modul konverter sebagai pengkonversi data satu bit ke delapan bit.

A. Perancangan Unit Kendali PLC

Dalam perancangan perangkat keras untuk PLC, penulis hanya menyusun dan merancang perangkat sensor dan aktuator yang akan digunakan. Untuk mempermudah dalam perancangan perangkat sensor dan aktuator langkah awal penulis harus menggambarkan terlebih dahulu *flow chart* dari tugas akhir penulis seperti yang terlihat pada gambar 3.1.



Gambar 3.1. Flow Chart 1 Sistem secara Keseluruhan

A.1 Perancangan Flow Chart Sistem secara Keseluruhan

Jika dilihat dari gambar 3.1, maka langkah awal dari flowchart yang dirancang adalah, program PLC di *load* setelah itu tekan TOMBOL Start/Stop sebagai tanda dimulainya pengontrolan motor. Sinyal terjadi jika TOMBOL Start/Stop ditekan, dan sinyal OFF terjadi jika TOMBOL ON/OFF dilepas. Saat TOMBOL Start/Stop ditekan, PLC langsung mengirim data 128 desimal kesemua output konversi yang dipakai, dengan maksud supaya perputaran motor langsung pada posisi setengah dari perputaran maksimum. Jika 255 desimal merupakan data maksimum sama dengan 10 volt pada tegangan maksimum inverter, maka 5 volt tegangan di inverter sama dengan data 128 desimal yang dikirim oleh PLC, rumus untuk mencari perbandingan antara desimal dengan tegangan adalah

sebagai berikut, misal untuk mencari nilai tegangan dari 128 desimal yang dikirim oleh PLC.

$$(10 : 255) \times 128 = 5 \qquad (3.1)$$

Setelah data 128 desimal terkirim keempat motor, maka masing-masing motor dapat langsung dikontrol kecepatan perputaran dan arah putar dengan menekan TOMBOL NAIK untuk menambah kecepatan, TOMBOL TURUN untuk mengurangi kecepatan, TOMBOL KIRI/KANAN untuk merubah arah putar motor kekiri atau kekanan. Sinyal kiri terjadi jika TOMBOL KIRI/KANAN dilepas dan sinyal kanan terjadi jika TOMBOL KIRI/KANAN ditekan. Serta TOMBOL ON/OFF untuk menjalankan dan mematikan motor. Dalam setiap penekanan TOMBOL NAIK/TURUN, akan menambah 4 desimal atau mengurangi 4 desimal data yang akan dikirim oleh PLC.

A.2 Perancangan Perangkat I/O PLC

Jika dilihat dari tabel 3.1 dan 3.2, maka jumlah tombol input dan output PLC yang akan digunakan penulis untuk disain sistem yang akan di rancang adalah sebagai berikut,

Untuk jumlah input ada tujuh belas yaitu:

a. Tombol ON / OFF bersifat tombol *detent*, 4 buah

b. Tombol Kiri / Kanan bersifat tombol detent, 4 buah

c. Tombol Turun bersifat tombol *push botton,* 4 buah

d. Tombol Naik bersifat tombol push botton, 4 buah

e. Tombol Start/Stop bersifat *detent,* 2 buah

Untuk jumlah output ada enam belas yaitu:

1. Lampu 1, untuk data yang dikirim pada output PLC 1

2. Lampu 2, untuk data yang dikirim pada output PLC 2
3. Lampu 3, untuk data yang dikirim pada output PLC 3
4. Lampu 4, untuk data yang dikirim pada output PLC 4
5. Output PLC untuk pilih gerak kiri motor 1
6. Output PLC untuk pilih gerak kanan motor 1
7. Output PLC untuk pilih gerak kiri motor 2
8. Output PLC untuk pilih gerak kanan motor 2
9. Output PLC untuk pilih gerak kiri motor 3
10. Output PLC untuk pilih gerak kanan motor 3
11. Output PLC untuk pilih gerak kiri motor 4
12. Output PLC untuk pilih gerak kanan motor 4

A.3 Perancangan Allocation List dan Address pada PLC

Allocation list adalah daftar yang berisi pemetaan sensor dan aktuator dengan alamat input dan output PLC, dengan adanya allocation ini sangat membantu dalam mengerjakan program dalam PLC, seperti yang terlihat pada tabel dibawah ini:

Tabel 3.1. Allocation List untuk Input

NO	Sensor / Aktuator	Nama Simbolis	Alamat PLC	Keterangan Sinyal
1	Detent Switch	T_KANAN1	I0.0	1 = Gerak Kanan 0 = Gerak Kiri
2	Detent Switch	T_ON1	I0.1	1 = ON 0 = OFF
3	Push Button Switch	T_NAIK1	I0.2	Tombol untuk menaikkan data motor
4	Push Button Switch	T_TURUN1	I0.3	Tombol untuk menurunkan data motor
5	Detent Switch	T_KANAN1	I0.4	1 = Gerak Kanan 0 = Gerak Kiri

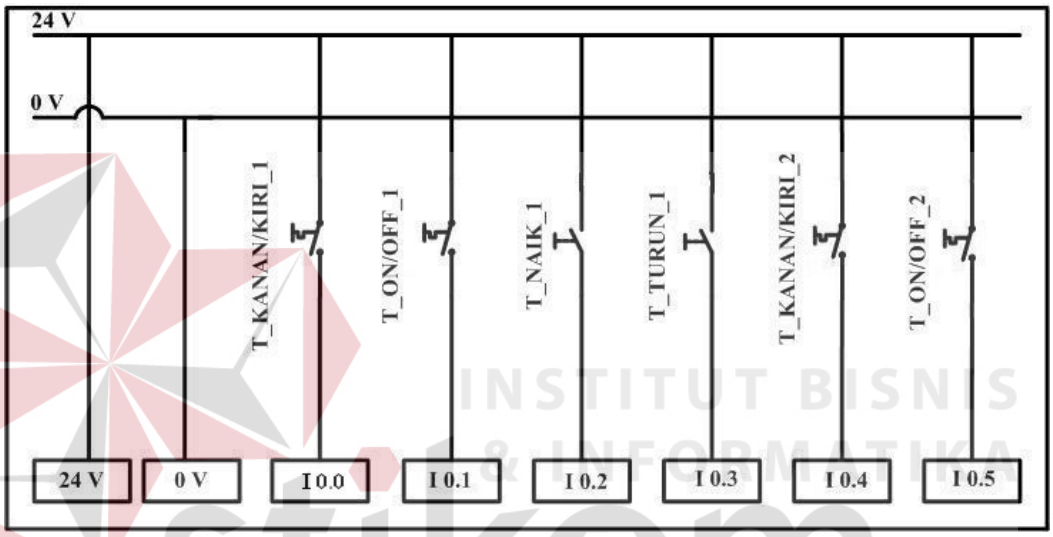
6	Detent Switch	T_ON2	I0.5	1 = ON 0 = OFF
7	Push Button Switch	T_NAIK2	I0.6	Tombol untuk menaikkan data motor
8	Push Button Switch	T_TURUN2	I0.7	Tombol untuk menurunkan data motor
9	Detent Switch	T_KANAN2	I1.0	1 = Gerak Kanan 0 = Gerak Kiri
10	Detent Switch	T_ON3	I1.1	1 = ON 0 = OFF
11	Push Button Switch	T_NAIK3	I1.2	Tombol untuk menaikkan data motor
12	Push Button Switch	T_TURUN3	I1.3	Tombol untuk menurunkan data motor
13	Detent Switch	T_KANAN3	I1.4	1 = Gerak Kanan 0 = Gerak Kiri
14	Detent Switch	T_ON4	I1.5	1 = ON 0 = OFF
15	Push Button Switch	T_NAIK4	I1.6	Tombol untuk menaikkan data motor
16	Push Button Switch	T_TURUN4	I1.7	Tombol untuk menurunkan data motor
17	Detent Switch	START	I2.0	1 = Start seluruh sistem PLC 0 = Stop seluruh sistem PLC

Tabel 3.2. Allocation List untuk Output

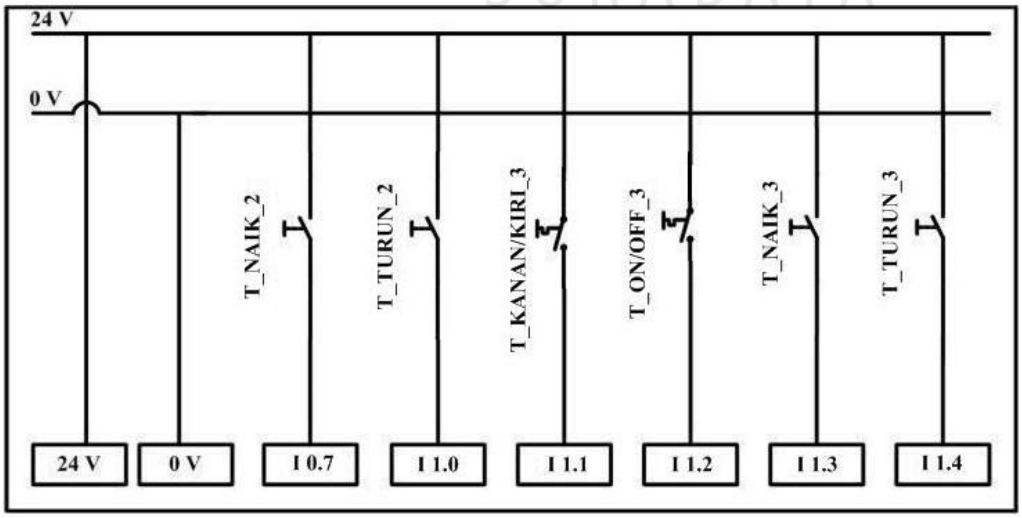
NO	Sensor / Aktuator	Nama Simbolis	Alamat PLC	Keterangan Sinyal
1	MOTOR DC 1	GERAKKIRI	O0.2	Motor bergerak kekiri
2	MOTOR DC 1	GERAKANAN	O0.3	Motor bergerak kekanan
3	MOTOR DC 2	GERAKKIRI	O0.4	Motor bergerak kekiri
4	MOTOR DC 2	GERAKANAN	O0.5	Motor bergerak kekanan
5	MOTOR DC 3	GERAKKIRI	O0.6	Motor bergerak kekiri
6	MOTOR DC 3	GERAKANAN	O0.7	Motor bergerak kekanan
7	MOTOR 3 FASA	GERAKKIRI	O1.0	Motor bergerak kekiri
8	MOTOR 3 FASA	GERAKANAN	O1.1	Motor bergerak kekanan
9	LAMPU 5	LHIDUP1	O1.2	Data ke output 1 PLC
10	LAMPU 6	LHIDUP2	O1.3	Data ke output 2 PLC
11	LAMPU 7	LHIDUP3	O1.4	Data ke output 3 PLC
12	LAMPU 8	LHIDUP4	O1.5	Data ke output 4 PLC

A.4 Perancangan Diagram Rangkaian Listrik.

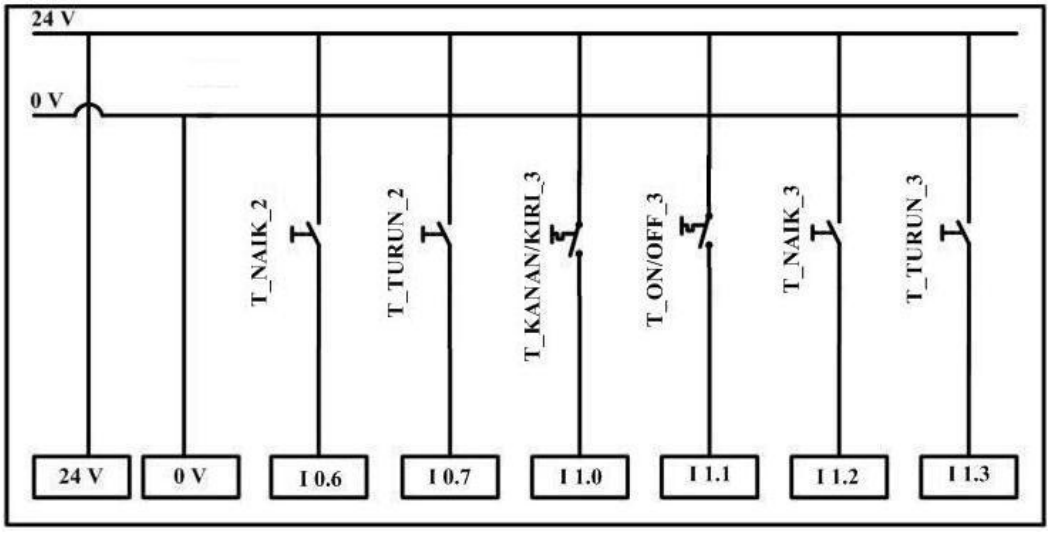
Diagram rangkaian listrik merupakan diagram yang menunjukkan hubungan antara catu daya, PLC, sensor dan aktuator. Diagram ini sangat berguna sebagai panduan pemasangan peralatan agar tidak terjadi hubungan singkat yang bisa merusak peralatan, seperti yang terlihat pada gambar 3.2a, 3.2b, 3.2c, 3.2d dan 3.2e.



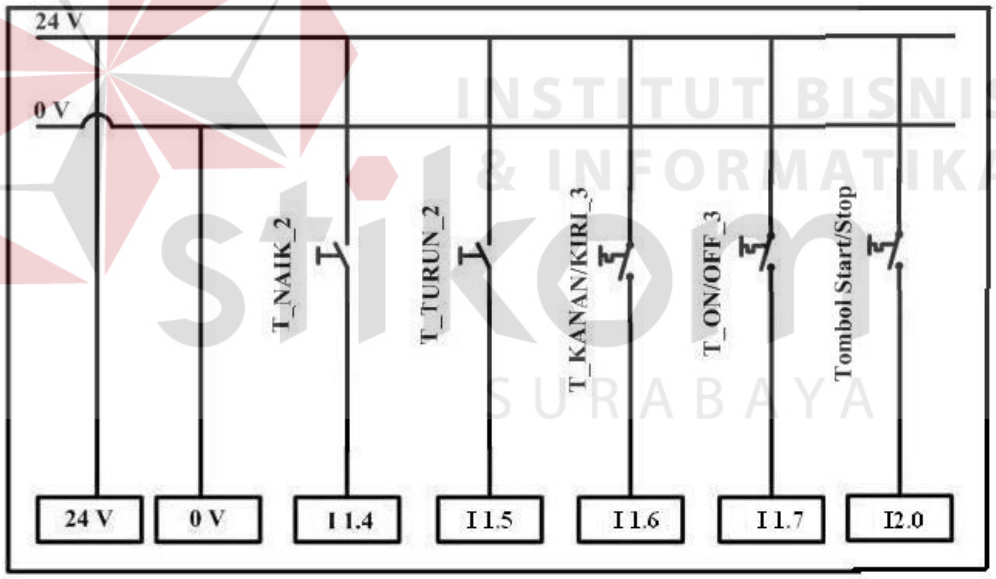
Gambar 3.2a. Diagram Rangkaian Listrik untuk Tombol pada PLC bagian 1



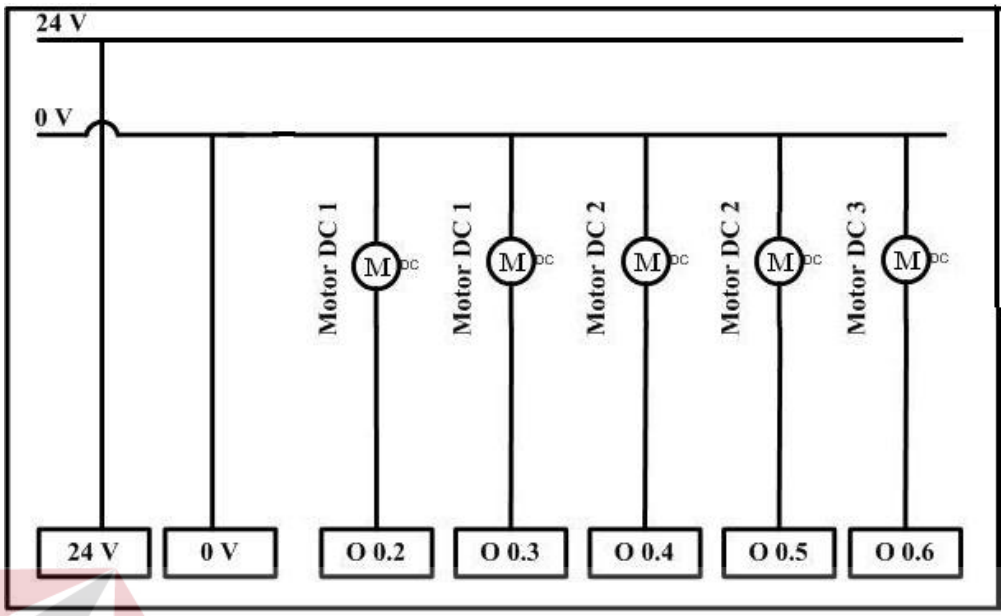
Gambar 3.2b. Diagram Rangkaian Listrik untuk Tombol pada PLC bagian 2



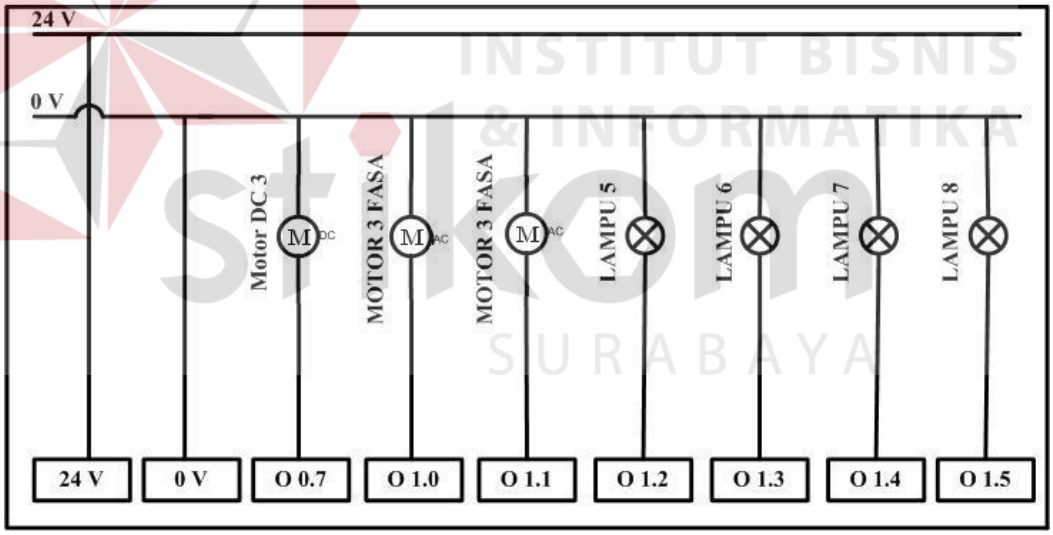
Gambar 3.2c. Diagram Rangkaian Listrik untuk Tombol pada PLC bagian 3



Gambar 3.2d. Diagram Rangkaian Listrik untuk Tombol pada PLC bagian 4



Gambar 3.2e. Diagram Rangkaian Listrik untuk Output pada PLC bagian 1



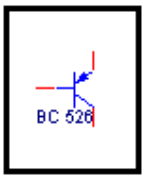
Gambar 3.2f. Diagram Rangkaian Listrik untuk Output pada PLC bagian 2

B. Perancangan Penurun Tegangan dan Tranfer Data

PLC memiliki output dengan jumlah 14 bit digital, yang masing-masing output memiliki tegangan 24 volt, sedangkan mikrokontroler mempunyai

tegangan maksimal 5 volt, dan transfer data dari output PLC dengan mikrokontroler harus memiliki kecepatan yang tinggi.

Maka untuk menurunkan tegangan dari 24 volt ke 5 volt serta memiliki kecepatan tranfer yang tinggi, penulis menggunakan transistor PNP bertipe BC 526, seperti yang terlihat ada gambar 3.3 dibawah ini.



Gambar 3.3. Lambang Transistor BC 526

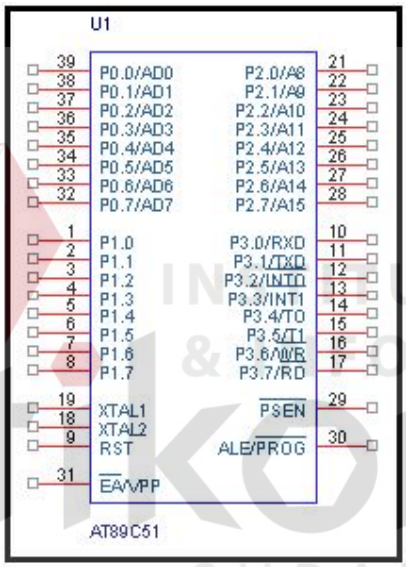
C. Perancangan Perangkat Keras Minimum Sistem 89C51

Minimum sistem 89C51 yang didisain oleh penulis berfungsi sebagai "konverter" (pengubah data delapan bit PLC yang ditransfer melalui satu bit output PLC, dan kemudian diterima oleh DAC delapan bit) setelah itu baru diterima oleh inverter motor tiga fasa. Penulis menggunakan mikrokontroler 89C51 karena dalam rancangan minimum sistem yang didisain penulis, membutuhkan 20 port I/O.

Jika menggunakan mikrokontroler 89C2051 tentu tidak cukup untuk port I/O nya, dengan demikian penulis menggunakan mikrokontroler 89C51 seperti pada gambar 3.4, sebagai minimum sistem pengkonversi. Berbagai macam kebutuhan port I/O tersebut adalah:

1. 4 port mikrokontroler yang dihubungkan dengan 4 jalur output PLC.
2. 4 port dihubungkan dengan dipswith.
3. 4 port untuk LE yang dihubungkan dengan laching.
4. 8 port untuk data yang dihubungkan dengan latching

Alasan lainnya kenapa penulis menggunakan mikrokontroler 89C51, karena jika sewaktu-waktu program yang dibuat oleh penulis tidak cukup ditampung oleh mikrokontroler 89C51 maka akan dengan mudah sekali diganti dengan mikrokontroler 89C52 yang memiliki memori dua kali 89C51, karena antara 89C51 dan 89C52 memiliki jumlah kaki pin yang sama, jadi tidak perlu merubah secara total minimum sistem yang telah dirancang, cukup IC nya saja yang diganti.

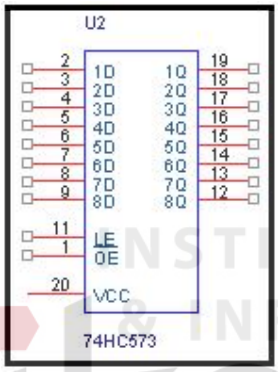


Gambar 3.4. AT89C51

D. Perancangan Perangkat Keras Latching

Karena motor yang dikontrol berjumlah empat buah dan bisa bergerak tanpa tergantung dengan yang lainnya maka data yang dikirim oleh PLC dan diteruskan oleh mikrokontroler harus di *Latch* (dipertahankan) untuk kemudian diteruskan kepada DAC yang berjumlah empat buah DAC sesuai dengan jumlah motor, IC Latch ini memiliki delapan bit data input yang mendapatkan data dari mikrokontroler, dan terdapat delapan bit output yang digunakan untuk mengontrol

motor tiga fasa, seperti yang terlihat pada gambar 3.5. Pin LE (*Lath Enable*) adalah *aktive high* sehingga dihubungkan dengan VCC, yang mempunyai fungsi sebagai meneruskan atau menahan data yang akan dikirim ke DAC. Jika LE aktif maka data 8 bit dari mikrokontroler diberi jalan untuk masuk ke IC latch, tapi jika LE tersebut tidak aktif maka data dari mikrokontroler tidak diberi jalan untuk masuk ke IC Latch. Sedangkan untuk pin OE (*Output Enable*) adalah aktif low sehingga harus diground, dan memepunyai fungsi sebagai pintu keluar bagi data 8 bit yang telah masuk ke IC Latc menuju ke DAC.



Gambar 3.5. IC Latch

E. Perancangan Perangkat Keras DAC

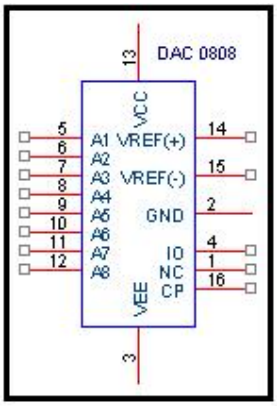
Dalam perancangan perangkat keras DAC, yang perlu diperhatikan pertama yaitu V_{REF} yang berfungsi sebagai tegangan referensi untuk DAC. Jika dilihat dari gambar 3.6, untuk mengendalikan motor tiga fasa membutuhkan perubahan tegangan dari 0 volt sampai 10 volt, maka V_{REF} pada DAC diberi catu daya sebesar 10 volt, dengan maksud supaya perubahan tegangan yang akan digunakan sebagai data pada DAC maksimum 10 volt, rumus untuk V_{REF} sebagai berikut:

$$R_{14} = V_{REF} / 0.002 \text{ A} \qquad (3.1)$$

R_{14} = Resistor yang terhung di kaki pin 14 DAC.

V_{REF} = Tegangan refensi 10 volt.

Maka, R_{14} = 10 V / 0.002A = 5000 ohm



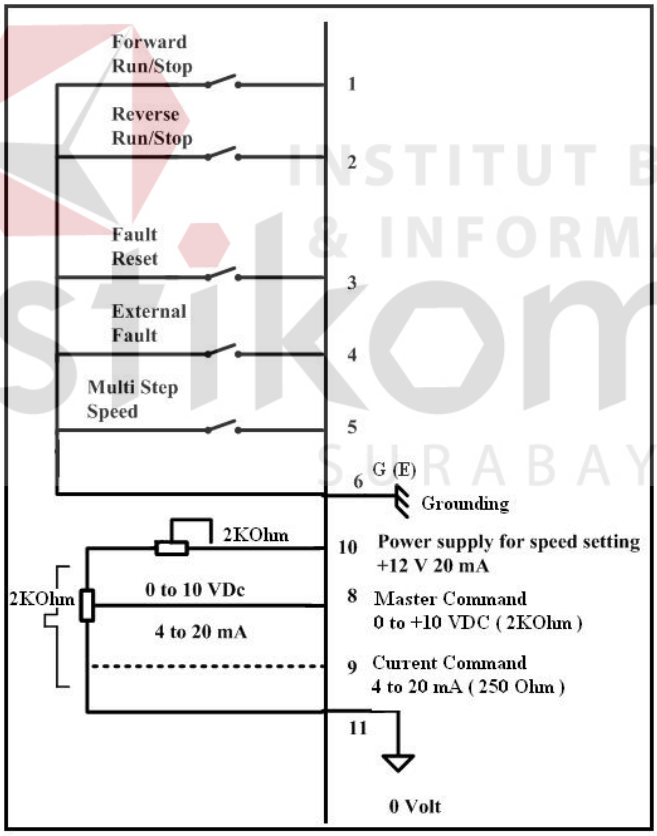
Gambar 3.6. DAC 0808

Dalam perancangan DAC karena output DAC masih berupa arus padahal yang dibutuhkan adalah berupa tegangan maka dibutuhkan rangkaian pendukung yang bisa merubah dari arus ke tegangan, untuk digunakanlah sebuah opamp untuk mengkonversi dari arus ke tegangan, dan jika dilihat gambar 3.6 pada bab pembuatan maka R_{14} nilainya harus sama dengan R di opamp, dengan maksud supaya konversi dari arus ke tegangan bisa tepat.

F. Perancangan Pemasangan Inverter Motor Tiga Fasa dengan PLC

Untuk mengontrol motor tiga fasa memerlukan sebuah inverter, inverter ini dihubungkan dengan motor tiga fasa, dan dihubungkan juga dengan PLC serta DAC, inverter yang terdapat di Laboratorium Kontrol, bertipe 3G3XV-A4002-E, seperti yang terlihat pada gambar 3.7 dan letak pin yang akan dihunbungkan gambar 3.8

Gambar 3.7. Inverter Motor Tiga Fasa



Gambar 3.8. Standart Wiring Diagram

Gamabar 3.9. Bentuk Motor Tiga Fasa

Gambar 3.7. merupakan bentuk dari inverter dan gambar 3.8 adalah posisi kaki pin pada inverter yang akan dihubungkan kepada PLC dan DAC, gambar 3.9 adalah motor tiga fasa.

Pada inverter kaki pin nomor 1 dan 2 berfungsi sebagai gerak kiri atau gerak kanan dari motor tiga fasa. Jika kaki pin nomor 1 aktif dan kaki pin 2 tidak aktif maka akan berputar ke kanan dan jika kaki pin 1 tidak aktif serta kaki pin 2 aktif maka akan bergerak kebalikannya. Kaki pin 1 dan 2 dihubungkan dengan *relay normally open* dan posisi saat sebelum *relay* tersambung harus dihubungkan dengan *ground* (nomor pin 6), sedangkan *koil*nya pada *relay* dihubungkan dengan 24 volt (Output PLC). Untuk mengontrol kecepatan motor tiga fasa terdapat pada kaki pin nomor 8, dengan kecepatan minimum 0 volt dan kecepatan maksimum 10 volt.

3.2.1 Perancangan Perangkat Lunak

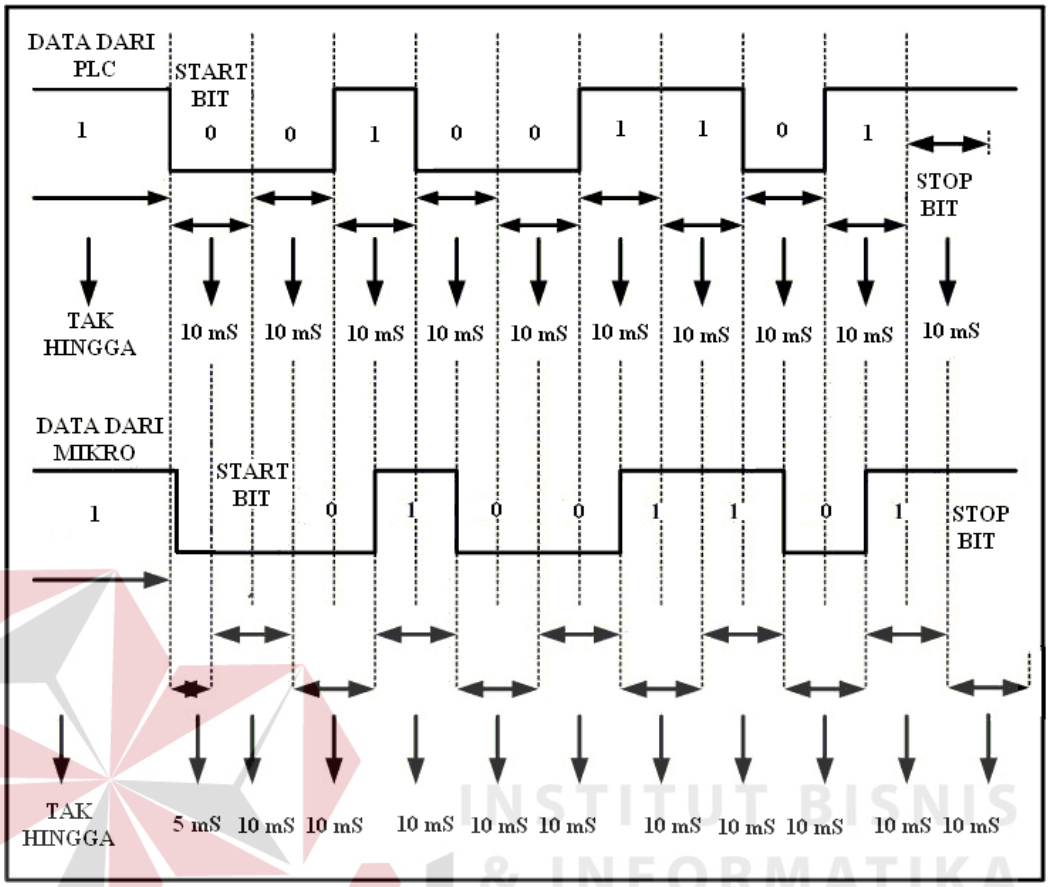
Untuk perancangan perangkat lunak, yaitu perangkat lunak yang digunakan untuk mengendalikan hardware. Dalam perancangan perangkat lunak

terdapat dua macam bahasa pemrograman yang digunakan oleh penulis. Yang pertama bahasa pemrograman STL yang merupakan bahasa pemrograman dari FESTO yang didisain khusus untuk PLC FESTO. Bahasa pemrograman yang kedua yaitu bahasa pemrograman yang digunakan untuk mikrokontroler 89C51 dengan menggunakan Bahasa *Assembly*. Semua Bahasa *Assembly* ditulis dengan menggunakan program Franklin Software (Proview32), Bahasa *Assembly* yaitu bahasa tingkat rendah, yang selanjutnya harus menggunakan bantuan rangkaian *writer* untuk mengisi mikrokontroler tersebut.

A. Perancangan Protokol Komunikasi Data

Sebelum membuat program STL dan assembly, sebaiknya merancang protokol komunikasi data antara PLC dan mikrokontroler. PLC mengirim data per bit (serial) sebanyak 8 bit lewat jalur output PLC. Protokol komunikasi data menggunakan mode *asinkron,* sehinngga dibutuhkan *start* bit dan *stop* bit, dan diterima oleh sebuah port pada mikroontroler.

Protokol pada PLC untuk mengirim sebuah *frame* data didisain sesuai dengan gambar 3.7. Mula-mula keadaan sinyal *high*, yang menandakan keadaan kosong (tidak ada transmisi data). *Start-bit* didisain *low* selama 10 mS, setelah keadaan kosong, begitu *start-bit* selesai berturut-turut dikirim 8-bit data dengan interval 10 mS untuk setiap bit. *Frame* ditutup dengan *stop-bit* yang didisain *high* selama 10 mS, setelah *stop-bit* selesai, jalur transmisi kembali pada keadaan kosong yang ditandai dengan sinyal *high*



Gambar 3.10. Protokol Komunikasi antara PLC dan Mikrokontroler

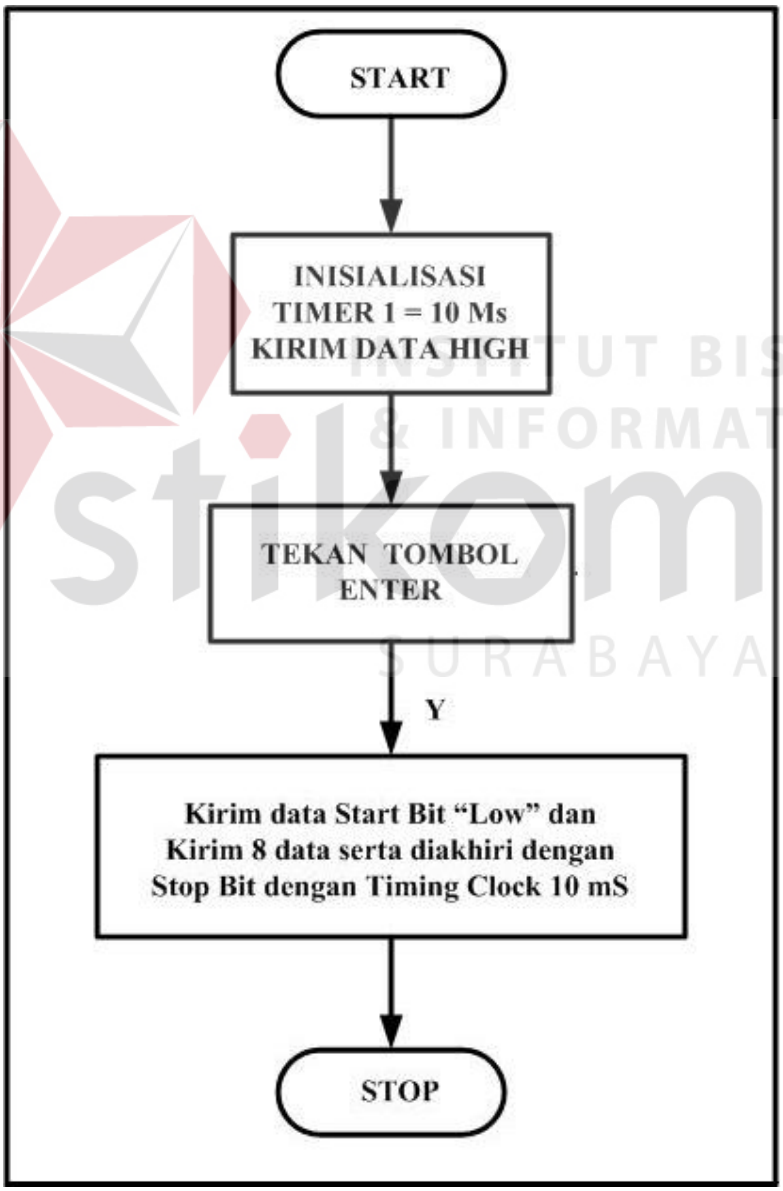
Jika dilihat gambar 3.10. selain sinyal dari PLC terdapat juga sinyal pada mikrokontroler, sinyal pada mikrokontroler berfungsi untuk menerima data dari PLC, tapi bagaimana caranya supaya sinyal dari PLC dapat diterima dengan baik oleh mikrokontroler, maka *start-bit* pada mikrokontroler memiliki interval 15 ms.

Pada saat PLC mengirim data *high* sebagai data kosong, maka mikrokontroler juga menerima sinyal *high*, begitu PLC megirim data *low* (*start-bit*), mikrokontroler juga *low*, tapi dengan interval 15 mS, dengan maksud supaya mikrokontroler bisa menangkap dengan baik 8 data berikutnya yang dikirim oleh PLC, sampai menerima *stop-bit high* yang menandakan berakhirnya 8 data yang

dikirim oleh PLC, begitu seterusnya jika terjadi pengiriman data dari PLC ke mikrokontroler.

B. Perancangan Komunikasi Data pada PLC

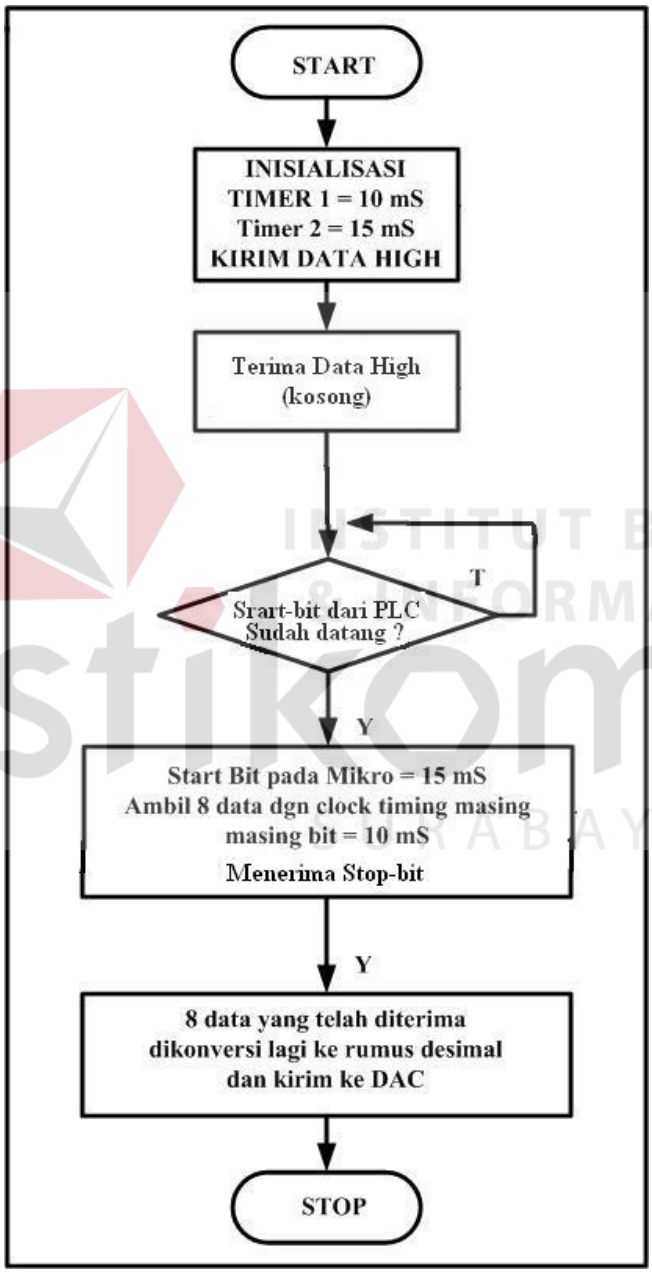
Dengan Flow Chart komunikasi data yang ada di PLC, yaitu bagaimana caranya PLC mengeluarkan sinyal sebelum start bit sampai ke stop bit dapat dilihat pada gambar 3.11.



Gambar 3.11. Flow Chart Program Kirim data dari PLC

C. Perancangan Komunikasi Data pada Mikrokontroler

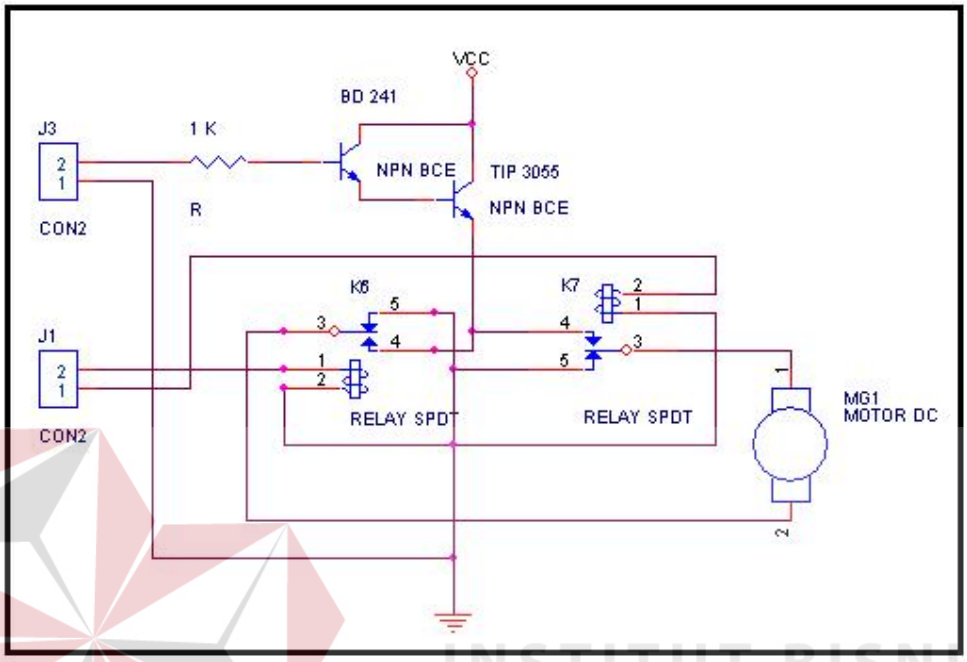
Dengan Flow Chart komunikasi data yang ada di mikrokontroler, yaitu bagaimana caranya mikrokontroler menerima dengan benar start bit, 8 data sampai ke stop bit dapat dilihat pada gambar 3.12.



Gambar 3.12. Flow Chart Program Terima data dari PLC

3.2 Pembuatan

3.2.1 Pembuatan Driver Motor



Gambar 3.13. Skematik Driver Motor

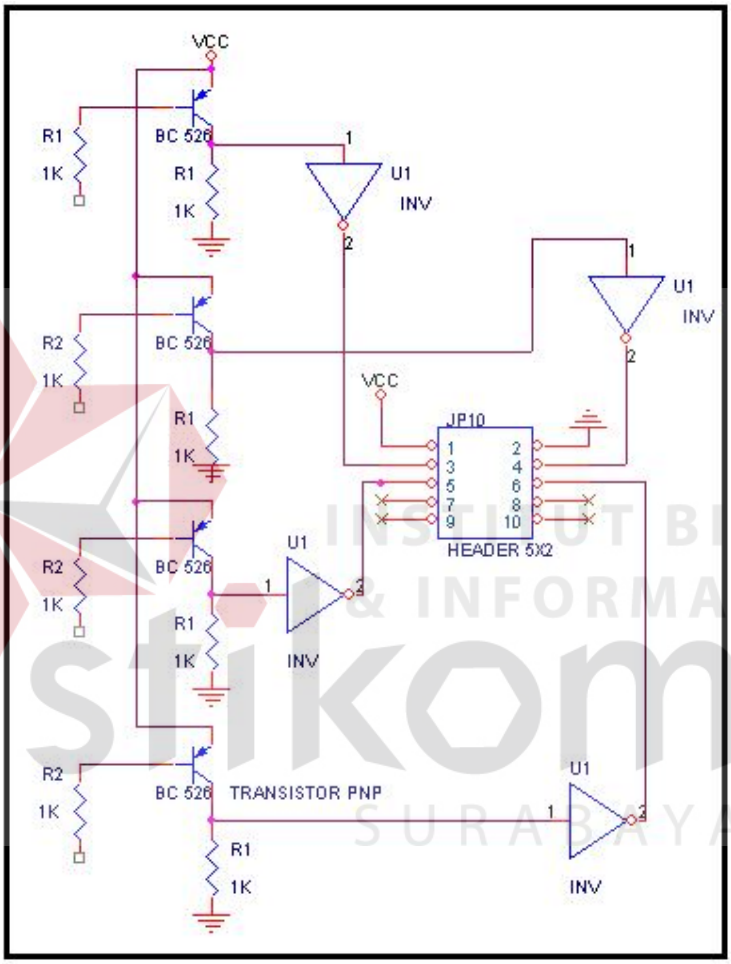
Dari gambar 3.13, terdapat 2 buah transistor berfungsi sebagai penguat arus, dengan maksud agar mampu memberikan arus yang yang dibutuhkan oleh motor DC. Digunakan 2 *relay* untuk mengatur arah putar motor (kiri atau kanan), dengan cara:

a. Menghubungkan 2 output PLC dengan *coil* tiap *relay*.
b. Menghubungkan output transistor penguat, dengan *contact-source* tiap *relay*.
c. Menghubungkan 2 penggerak motor dengan *contact normally Open* tiap *relay*

3.2.2 Pembuatan Penurun Tegangan dan Tranfer Data

PLC dan mikrokontroler saling berkomunikasi satu arah, dan untuk berkomunikasi antara PLC dengan mikrokontroler tidak bisa langsung

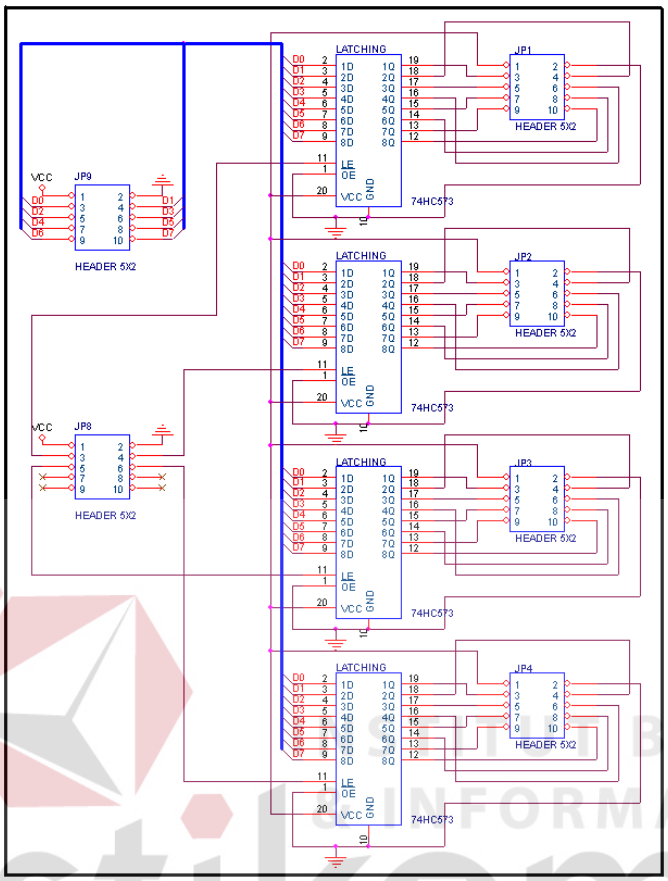
dihubungkan karena output PLC bertegangan 24 volt, sedangkan mikrokontroler membutuhkan 5 volt, untuk itu dibutuhkan penurun tegangan, seperti yang terlihat pada gambar 3.14.



Gambar 3.14. Skematik Penurun tegangan dan Transfer data

Dari gambar 3.14, terlihat terdapat transistor BC 526, yang berfungsi menurunkan tegangan dari 24 volt ke 5 volt, jika pada PLC mengeluarkan sinyal *high* pada kaki kolektor transistor mengeluarkan sinyal *low*, maka harus diberi inverter supaya jika PLC memberi sinyal *high*, keluaran kolektor akan *high* juga dan baru masuk ke mikrokontroler.

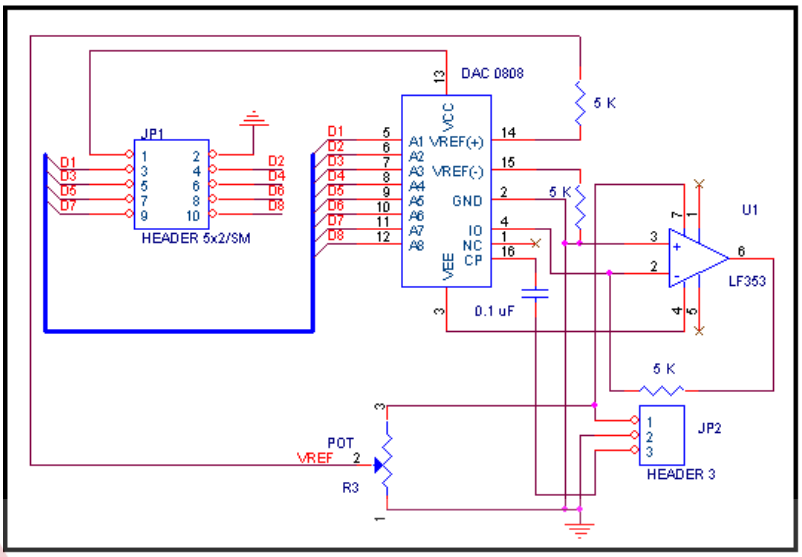
3.2.3 Pembuatan Perangkat Keras Latching



Gambar 3.15. Skematik latching

Dari gambar 3.15. dapat diterangkan bahwa latching akan menampung dan menyimpan data yang akan dikirim oleh mikrokontroler, dengan adanya latching maka data 8 bit yang dikirim oleh mikrokontroler dapat di teruskan ke empat motor, dan data berakhir yang dikirim dapat disimpan untuk mempertahankan perecepatan motor serta tidak saling tergantung antara motor satu dengan motor yang lainnya.

3.2.4 Pembuatan Rangkaian DAC



Gambar 3.15. Skematik DAC

DAC akan merubah data digital yang dikirim oleh mikrokontroler menjadi analog, dirubah ke analog karena inverter motor tiga fasa membutuhkan data analog untuk mengatur kecepatan motor yang berupa tegangan antara 0 sampai 10 volt. Untuk gerak kiri dan kanannya tidak perlu DAC, tapi langsung dihungkan ke PLC, karena hanya berupa *relay*

DAC 0808 memiliki resolusi 16 sampai 18 bit, hampir semuanya bersifat *monotonik* dengan kesalahan kurang dari. ½ LSB pada setiap tingkat keluaran. DAC 0808 merupakan konverter D/A tangga R-2R 8-bit yang dilengkapi dengan sumber arus acuan dan delapan buah transistor saklar untuk mengarahkan arus biner. Suatu tegangan dan hambatan eksternal dipergunakan untuk mengatur arus acuan pada nilai yang lazim berlaku yaitu 2mA, dengan waktu pemantapan 150 ns.

Cara kerja Rangkaian DAC adalah sebagai berikut:

1. Kaki pin 2 ditanahkan.

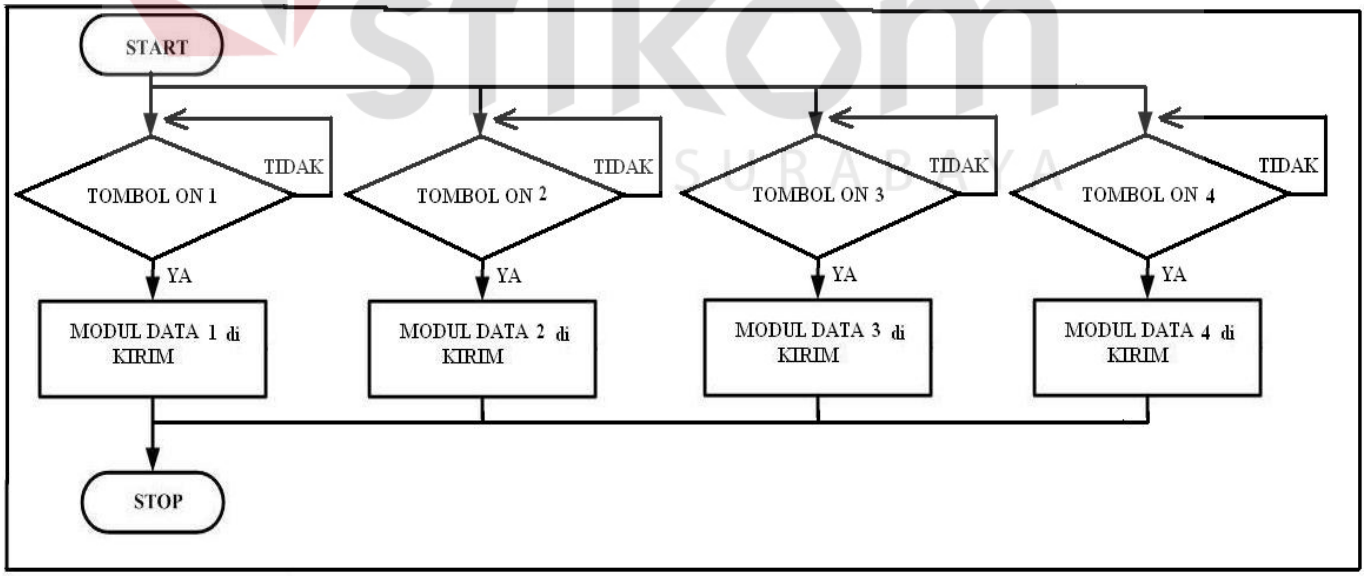
2. V_{EE} pada kaki pin 3 harus dipasang pada tegangan -15 Volt.
3. Kaki pin 4, adalah saluran balik dari tanah *(ground)*, bagi arus yang keluar dari rangkaian tangga dan biasanya dihubungkan dengan op-amp, op-amp berfungsi sebagai penguat amplitudo.
4. Kaki pin 5 sampai 12 merupakan saluran bagi 8 bit data masukan.
5. Kaki pin 13 harus dipasang pada caru daya +15 Volt.
6. Kaki pin 14 dihubungkan dengan catu daya tegangan positif melalui sebuah hmbatan R_{14} yang berupa potensiometer trimer, memungkinkan untuk menetapkan arus acuan (I_{REF}) pada harga 2mA.
7. Kaki pin 15 ditanahkan melalui sebuah hambatan R_{15} dan berukuran sama dengan R_{14}, hambatan ini mengkompensasi pergeseran (*drift*) tahap masukan konverter. Jika diperhatikan bahwa I_{OUT} menggerakkan masukan membalik dari sebuah op-amp LF353.
8. Sebuah kapasitor antara kaki pin 16 dan kaki pin 13 berfungsi untuk memberi kompensasi frekuensi bagi piranti ini.

3.2.5 Pembuatan Program Sederhana Kirim Data pada Program PLC

```
STEP INIT
     THEN    LOAD     V1  “10 mS
             TO      TP0
             SET L_HIDUP  “Data awal sebagai tanda untuk mikro bahwa
     bukan
                          “data yang perlu diambil
STEP TOMBOL
     IF      T_ENTER
     THEN    NOP
STEP CEK
     IF      T_ENTER
     THEN    RESET L_HIDUP  “START BIT
             SET T0         “10 Ms
STEP
     IF      N       T0     “Pengecekan timer sudah habis belum
     THEN    SET L_HIDUP    “Data 1 = 1
             SET T0         “Timer 0 sebagai clock timing 10 mS
STEP
     IF      N       T0     “Pengecekan timer sudah habis belum
     THEN    SET L_HIDUP    “Data 2 = 1
             SET T0         “Timer 0 sebagai clock timing 10 mS
```

```
STEP
      IF      N       T0      “Pengecekan timer sudah habis belum
      THEN    RESET L_HIDUP   “Data 3 = 0
              SET T0          “Timer 0 sebagai clock timing 10 mS
STEP
      IF      N       T0      “Pengecekan timer sudah habis belum
      THEN    SET L_HIDUP     “Data 4 = 1
              SET T0          “Timer 0 sebagai clock timing 10 mS
STEP
      IF      N       T0      “Pengecekan timer sudah habis belum
      THEN    RESET L_HIDUP   “Data 5 = 0
              SET T0          “Timer 0 sebagai clock timing 10 mS
STEP
      IF      N       T0      “Pengecekan timer sudah habis belum
      THEN    RESET L_HIDUP   “Data 6 = 0
              SET T0          “Timer 0 sebagai clock timing 10 mS
STEP
      IF      N       T0      “Pengecekan timer sudah habis belum
      THEN    SET L_HIDUP     “Data 7 = 1
              SET T0          “Timer 0 sebagai clock timing 10 mS
STEP
      IF      N       T0      “Pengecekan timer sudah habis belum
      THEN    SET L_HIDUP     “Data 8 = 1
              SET T0          “Timer 0 sebagai clock timing 10 mS
STEP
      IF      N       T0      “Pengecekan timer sudah habis belum
      THEN    SET L_HIDUP     “Data 9 = STOP BIT
              SET T0          “Timer 0 sebagai clock timing 10 mS
STEP
      IF      N       T0      “Pengecekan timer sudah habis belum
      THEN    JMP TO INIT
```

Untuk pengiriman data secara lengkap program diatas tadi dijadikan suatu modul, dan modul tersebut akan dieksekusi jika telah ditekan TOMBOL ON/OFF, seperti pada gambar berikut



Gambar 3.16 Flowchart Modul Data

3.2.6 Pembuatan Program Terima 4 Data pada Mikrokontroler

```
ORG 100H
LJMP MULAI

COUNTER_5MS EQU 80H

HITUNG_TIMER_1:
            MOV COUNTER_5MS,#02; 5 DIKALI 2 JADI 10 MILLI S = 1 detik

TUNGGU_500MS_1:
            ACALL DELAY_5MS_1
            DJNZ COUNTER_5MS,TUNGGU_500MS_1
RET

DELAY_5MS_1:
            PUSH TMOD
            MOV TMOD,#21H
            MOV TH0,#0EDH
            MOV TL0,#0FFH
            SETB TR0

TUNGGU_5MS_1:
            JBC TF0,SUDAH_5MS_1
            AJMP TUNGGU_5MS_1

SUDAH_5MS_1:
            CLR TR0
            POP TMOD
RET

HITUNG_TIMER_2:
            MOV COUNTER_5MS,#03 ; 5 DIKALI 3 JADI 15 MILLI S

TUNGGU_500MS_2:
            ACALL DELAY_5MS_2
            DJNZ COUNTER_5MS,TUNGGU_500MS_2
RET

DELAY_5MS_2:
            PUSH TMOD
            MOV TMOD,#21H
            MOV TH0,#0EDH
            MOV TL0,#0FFH
            SETB TR0

TUNGGU_5MS_2:
            JBC TF0,SUDAH_5MS_2
            AJMP TUNGGU_5MS_2

SUDAH_5MS_2:
            CLR TR0
            POP TMOD
RET

MULAI:
            MOV P1,#00H
            MOV P2,#0FFH         ;LE = 1
            MOV P2,#00H          ;LE = 0

            MOV P1,#0ffh
            MOV P3,#0ffH
START:
      MOV R0,#00H
                  MOV R1,#00H
                  MOV R2,#00H
                  MOV R3,#00H
                  MOV R4,#00H
                  MOV R5,#00H
                  MOV R6,#00H
                  MOV R7,#00H
CEK1:
            JB P3.0,CEK2    ;OUTPUT BIT 1 DR PLC, P3.0=1 MK KE CEK2, JK
                            ;P3.0=0 KE STARTBIT
```

```
            JB P0.0,STARTBIT1


CEK2:
            JB P3.1,CEK3    ;OUTPUT BIT 2 DR PLC, P3.1=1 MK KE CEK3, JK
                            ;P3.0=0 KE STARTBIT
            JB P0.1,STARTBIT2


CEK3:
            JB P3.2,CEK4    ;OUTPUT BIT 2 DR PLC, P3.2=1 MK KE CEK4, JK
                            ;P3.0=0 KE STARTBIT
            JB P0.2,STARTBIT31


CEK4:
            JB P3.3,CEK1    ;OUTPUT BIT 2 DR PLC, P3.3=1 MK KE CEK1, JK
                            ;P3.0=0 KE STARTBIT
            JB P0.3,STARTBIT41
            JMP CEK1
STARTBIT41:
            JMP STARTBIT4
STARTBIT31:
            JMP STARTBIT3


STARTBIT1:
            ACALL HITUNG_TIMER_2                ;DATA START BIT 15 MS
MOT11:      JNB P3.0,MOT12
            MOV R0,#01H           ;DATA 1  10 MS
MOT12:      ACALL HITUNG_TIMER_1
            JNB P3.0,MOT13
            MOV R1,#02H           ;DATA 2
MOT13:      ACALL HITUNG_TIMER_1
            JNB P3.0,MOT14
            MOV R2,#04H                         ;DATA 3
MOT14:      ACALL HITUNG_TIMER_1
            JNB P3.0,MOT15
            MOV R3,#08H                         ;DATA 4
MOT15:      ACALL HITUNG_TIMER_1
            JNB P3.0,MOT16
            MOV R4,#10H                         ;DATA 5
MOT16:      ACALL HITUNG_TIMER_1
            JNB P3.0,MOT17
            MOV R5,#20H                         ;DATA 6
MOT17:      ACALL HITUNG_TIMER_1
            JNB P3.0,MOT18
            MOV R6,#40H                                 ;DATA 7
MOT18:      ACALL HITUNG_TIMER_1
            JNB P3.0,MOT19
            MOV R7,#80H                         ;DATA 8
MOT19:      ACALL HITUNG_TIMER_1
            MOV 41,#01H
TAHANDULU1:
            JB P3.0,MULAI11      ;JIKA P3.0 BIT YG KE 9 = 0 MAKA DITAHAN
                                  ;DULU SAMPAI P3.0 = 1 KE MULAI1
            JMP TAHANDULU1
MULAI11:
            JMP MULAI1

STARTBIT2:
            ACALL HITUNG_TIMER_2                ;DATA START BIT 15 MS
MOT21:      JNB P3.1,MOT22
            MOV R0,#01H                         ;DATA 1  10 MS
MOT22:      ACALL HITUNG_TIMER_1
            JNB P3.1,MOT23
            MOV R1,#02H                         ;DATA
MOT23:      ACALL HITUNG_TIMER_1
            JNB P3.1,MOT24
            MOV R2,#04H                         ;DATA 3
MOT24:      ACALL HITUNG_TIMER_1
            JNB P3.1,MOT25
            MOV R3,#08H                         ;DATA 4
MOT25:      ACALL HITUNG_TIMER_1
            JNB P3.1,MOT26
```

```
                MOV R4,#10H                         ;DATA 5
MOT26:          ACALL HITUNG_TIMER_1
                JNB P3.1,MOT27
                MOV R5,#20H                         ;DATA 6
MOT27:          ACALL HITUNG_TIMER_1
                JNB P3.1,MOT28
                MOV R6,#40H                         ;DATA 7
MOT28:          ACALL HITUNG_TIMER_1
                JNB P3.1,MOT29
                MOV R7,#80H                         ;DATA 8
MOT29:          ACALL HITUNG_TIMER_1
                MOV 40,#01H
TAHANDULU2:
                JB P3.1,MULAI12      ;JIKA P3.0 BIT YG KE 9 = 0 MAKA DITAHAN
                                      ;DULU SAMPAI P3.0 = 1 KE MULAI1
                JMP TAHANDULU2
MULAI12:
                JMP MULAI1

STARTBIT3:
                ACALL HITUNG_TIMER_2                ;DATA START BIT 15 MS
MOT31:          JNB P3.2,MOT32
                MOV R0,#01H                         ;DATA 1  10 MS
MOT32:          ACALL HITUNG_TIMER_1
                JNB P3.2,MOT33
                MOV R1,#02H                         ;DATA
MOT33:          ACALL HITUNG_TIMER_1
                JNB P3.2,MOT34
                MOV R2,#04H                         ;DATA 3
MOT34:          ACALL HITUNG_TIMER_1
                JNB P3.2,MOT35
                MOV R3,#08H                         ;DATA 4
MOT35:          ACALL HITUNG_TIMER_1
                JNB P3.2,MOT36
                MOV R4,#10H                         ;DATA 5
MOT36:          ACALL HITUNG_TIMER_1
                JNB P3.2,MOT37
                MOV R5,#20H                         ;DATA 6
MOT37:          ACALL HITUNG_TIMER_1
                JNB P3.2,MOT38
                MOV R6,#40H                         ;DATA 7
MOT38:          ACALL HITUNG_TIMER_1
                JNB P3.2,MOT39
                MOV R7,#80H                         ;DATA 8
MOT39:          ACALL HITUNG_TIMER_1
                MOV 39,#01H
TAHANDULU3:
                JB P3.2,MULAI13          ;JIKA P3.0 BIT YG KE 9 = 0 MAKA
DITAHAN
                                         ;DULU SAMPAI P3.0 = 1 KE MULAI1
                JMP TAHANDULU3
MULAI13:
                JMP MULAI1

STARTBIT4:
                ACALL HITUNG_TIMER_2                ;DATA START BIT 15 MS
MOT41:          JNB P3.3,MOT42
                MOV R0,#01H                         ;DATA 1  10 MS
MOT42:          ACALL HITUNG_TIMER_1
                JNB P3.3,MOT43
                MOV R1,#02H                         ;DATA
MOT43:          ACALL HITUNG_TIMER_1
                JNB P3.3,MOT44
                MOV R2,#04H                         ;DATA 3
MOT44:          ACALL HITUNG_TIMER_1
                JNB P3.3,MOT45
                MOV R3,#08H                         ;DATA 4
MOT45:          ACALL HITUNG_TIMER_1
                JNB P3.3,MOT46
                MOV R4,#10H                         ;DATA 5
MOT46:          ACALL HITUNG_TIMER_1
                JNB P3.3,MOT47
                MOV R5,#20H                         ;DATA 6
MOT47:          ACALL HITUNG_TIMER_1
                JNB P3.3,MOT48
```

```
                MOV R6,#40H                             ;DATA 7
MOT48:          ACALL HITUNG_TIMER_1
                JNB P3.3,MOT49
                MOV R7,#80H                             ;DATA 8
MOT49:          ACALL HITUNG_TIMER_1
                MOV 38,#01H

TAHANDULU4:
                JB P3.3,MULAI1        ;JIKA P3.0 BIT YG KE 9 = 0 MAKA DITAHAN
                                       ;DULU SAMPAI P3.0 = 1 KE MULAI1
                JMP TAHANDULU4

MULAI1:
                MOV A,R0    ;MENGELUARKAN DATANYA KE PORT 1 DIMIKRO
                ADD A,R1
                ADD A,R2
                ADD A,R3
                ADD A,R4
                ADD A,R5
                ADD A,R6
                ADD A,R7
                MOV P1,A

LETH1:
                MOV A,41
                CJNE A,#01H,LETH2
                MOV P2,#04H             ;LE = 1
                LCALL HITUNG_TIMER_2
                LCALL HITUNG_TIMER_2
                LCALL HITUNG_TIMER_2
                MOV P2,#00H
                MOV 41,#00H

LETH2:
                MOV A,40
                CJNE A,#01H,LETH3
                MOV P2,#02H             ;LE = 2
                LCALL HITUNG_TIMER_2
                LCALL HITUNG_TIMER_2
                LCALL HITUNG_TIMER_2
                MOV P2,#00H
                MOV 40,#00H

LETH3:
                MOV A,39
                CJNE A,#01H,LETH4
                MOV P2,#40H             ;LE = 3
                LCALL HITUNG_TIMER_2
                LCALL HITUNG_TIMER_2
                LCALL HITUNG_TIMER_2
                MOV P2,#00H
                MOV 39,#00H

LETH4:
                MOV A,38
                CJNE A,#01H,LETH5
                MOV P2,#20H             ;LE = 4
                LCALL HITUNG_TIMER_2
                LCALL HITUNG_TIMER_2
                LCALL HITUNG_TIMER_2
                MOV P2,#00H
                MOV 38,#00H
LETH5:
  LJMP START
END
```