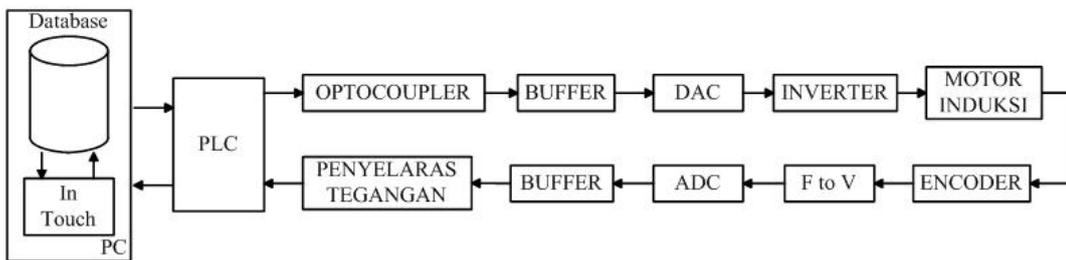


### 3. PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai perencanaan sistem yang digunakan pada Implementasi InSQL Database Server Pada Sistem SCADA. Secara umum, blok diagram dari sistem kerja pengendalian kecepatan motor dapat dilihat pada gambar 3.1. di bawah ini.



Gambar 3.1. Blok Diagram Sistem Detail

Cara kerja dari sistem ini adalah pada PC akan ada tampilan *window* dengan menggunakan *Wonderware InTouch*, dari *window* tersebut dapat mengatur kecepatan motor yang diinginkan berdasarkan *set point* yang ada atau dapat juga membuat *set point* yang baru. Nilai *set point* akan mengontrol kecepatan motor yang diinginkan, lalu data kecepatan akan dilewatkan melalui PLC. Keluaran digital dari PLC akan dikonversi oleh DAC menjadi analog yang berupa tegangan DC, lalu *inverter* akan mengatur frekuensi dan kecepatan motor AC. *Encoder* disini berfungsi untuk mengukur kecepatan motor, lalu output dari *encoder* yang berupa pulsa akan diubah dahulu dalam bentuk tegangan dengan menggunakan rangkaian F to V, yang kemudian akan dihubungkan dengan rangkaian ADC sehingga dapat masuk ke dalam PLC dan akan ditampilkan melalui PC. Perencanaan ini meliputi perencanaan *hardware* dan perencanaan *software*. *Software* yang dibuat adalah *software* yang bertujuan untuk menjalankan plant.

### 3.1. Perencanaan Hardware

Pada perencanaan *hardware* ini akan dibagi menjadi beberapa bagian, diantaranya meliputi:

- Desain *Hardware Output* :
  - *Optocoupler*
  - *Buffer*
  - DAC (*Digital to Analog Converter*)
- *Inverter altivar*
- Motor AC 3 phasa
- Desain *Hardware Input* :
  - *Encoder*
  - F to V
  - ADC (*Analog to Digital Converter*)
  - *Buffer*
  - Penyelaras Tegangan (*Comparator*)

#### 3.1.1. Desain Hardware Output

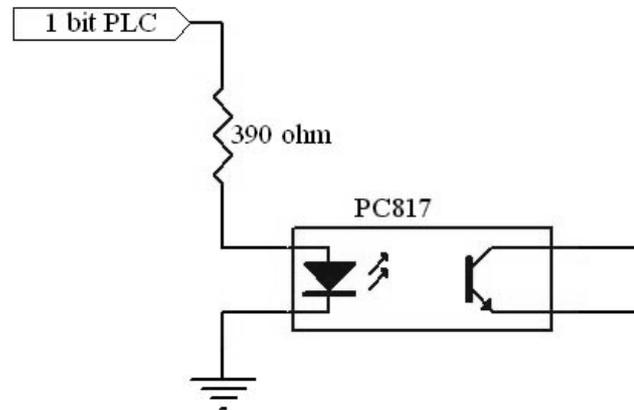
##### 3.1.1.1. Optocoupler

*Optocoupler* digunakan untuk menyesuaikan kondisi antara PLC dan DAC dan memisah *ground* antara *ground* PLC dengan *ground* DAC 0808. Digunakan rangkaian *optocoupler* karena pada saat penggabungan, saat *ground* dari *supply altivar* dihubungkan dengan *ground* F to V menghasilkan nilai yang tidak diinginkan. Pada saat *supply inverter* diberi tegangan 0 volt, outputan dari F to V menjadi 6 volt, padahal seharusnya pada saat *supply inverter* diberi tegangan 0 volt, outputan F to V juga harus 0 volt. PLC yang digunakan memiliki *output relay*. Dengan demikian ada saat di mana PLC tidak memberikan *logic* yang pasti yaitu:

- a. “0” dan *High Impedance* jika PLC memiliki karakteristik “*output sinking*”
- b. atau “1” dan *High Impedance* jika PLC memiliki “*output sourcing*”.

PLC OMRON CPM1 memiliki karakteristik yang memberikan kondisi *output sourcing*, sehingga kondisi yang terjadi adalah “1” dan *High Impedance*. Dengan demikian akan timbul masalah baru, karena kondisi *High Impedance*

selalu dibaca sebagai *logic* “1” pada *level* tegangan TTL di mana DAC bekerja. Untuk itu dibuat sebuah rangkaian pengkondisi sinyal yang bisa merubah *High Impedance* sebagai *logic* “0”.



Gambar 3.2. Desain *Optocoupler-1*

Input :

$$\text{Diketetahui : } V_{\text{led}} = 1 \text{ V}$$

$$I_{\text{led}} = 10 \text{ mA}$$

Sumber: Sharp Corporation. *Sharp Device Specification for Photocoupler PC817*. 25 Maret 2005. <<http://www.alldatasheet.com/datasheet-pdf/pdf/77784/AUK/SPC817M.html>>.

$$I_{\text{bit PLC}} = 5 \text{ V (pada logic "1" dan COM output PLC diberi tegangan 5V)}$$

Ditanya :  $R$

Jawab :

$$V_R = V_{\text{input}} - V_{\text{led}}$$

$$= 5 - 1$$

$$= 4 \text{ V}$$

$$R = \frac{V_R}{I_R} = \frac{4V}{10mA} = 400 \text{ ohm} \approx 390 \text{ ohm}$$

Output:

$$\text{Diketetahui : } I = 4 - 30 \text{ mA, } V_{CE} = 0,2 \text{ V}$$

Sumber: Sharp Corporation. *Sharp Device Specification for Photocoupler PC817*. 25 Maret 2005. <<http://www.alldatasheet.com/datasheet-pdf/pdf/77784/AUK/SPC817M.html>>.

Ditanya :  $R?$

Jawab :

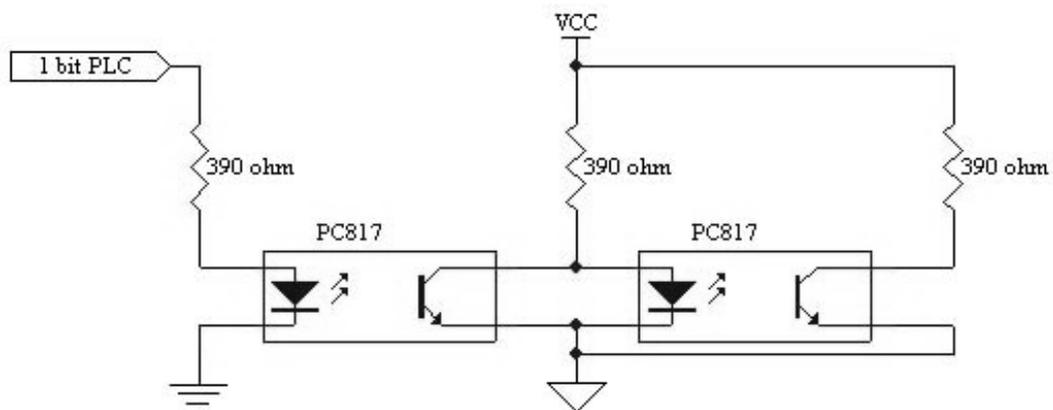
$$\begin{aligned} V_R &= V_{input} - V_{CE} \\ &= 5 - 0,2 \\ &= 4,8 \text{ V} \end{aligned}$$

$$R_{\max} = \frac{V_R}{I_R} = \frac{4,8V}{4mA} = 1.2 \text{ kohm}$$

$$R_{\min} = \frac{V_R}{I_R} = \frac{4,8V}{30mA} = 160 \text{ ohm}$$

Dari perhitungan tersebut kemudian diambil R yang berada di range tersebut yaitu sebesar 390 ohm.

Rangkaian tersebut akan memberikan 2 kondisi, saat PLC ber-logic "1", maka *optocoupler* memberikan logic "0". Sedangkan pada saat PLC *High Impedance*, opto coupler memberikan logic "1". Namun kondisi ini belum menjawab kebutuhan. Oleh karena itu hasil dari pengkondisian sinyal tersebut kembali dimasukkan pada pengkondisian sinyal yang sama seperti gambar 3.2.



Gambar 3.3. Desain *Optocoupler-2*

Input optocoupler2:

Diketahui :  $I = 4 - 50 \text{ mA}$ ;  $V = 1 \text{ V}$

Sumber: Sharp Corporation. *Sharp Device Specification for Photocoupler PC817*. 25 Maret 2005. <<http://www.alldatasheet.com/datasheet-pdf/pdf/77784/AUK/SPC817M.html>>.

Ditanya :  $R?$

Jawab :

$$V_R = V_{input} - V_{led}$$

$$= 5 - 1$$

$$= 4 \text{ V}$$

$$R_{\max} = \frac{V_R}{I_R} = \frac{4,8V}{4mA} = 1.2 \text{ kohm}$$

$$R_{\min} = \frac{V_R}{I_R} = \frac{4,8V}{30mA} = 160 \text{ ohm}$$

Karena *input optocoupler 2* terkoneksi dengan *optocoupler 1* maka R yang terpasang adalah 390 ohm.

Output optocoupler2:

$$\text{Diketahui : } I = 4 - 30 \text{ mA, } V = 0,2 \text{ V}$$

Sumber: Sharp Corporation. *Sharp Device Specification for Photocoupler PC817*.  
25 Maret 2005. <<http://www.alldatasheet.com/datasheet-pdf/pdf/77784/AUK/SPC817M.html>>

Ditanya :  $R?$

Jawab :

$$V_R = V_{input} - V_{led}$$

$$= 5 - 0,2$$

$$= 4,8 \text{ V}$$

$$R_{\max} = \frac{V_R}{I_R} = \frac{4,8V}{4mA} = 1.2 \text{ kohm}$$

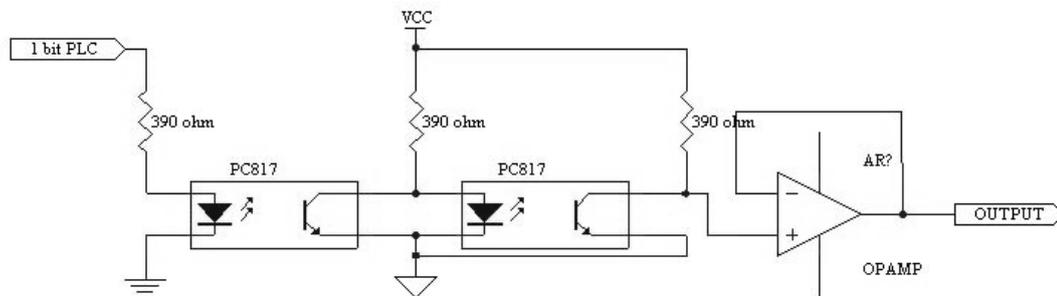
$$R_{\min} = \frac{V_R}{I_R} = \frac{4,8V}{30mA} = 160 \text{ ohm}$$

Dari perhitungan tersebut kemudian diambil R yang berada di range tersebut yaitu sebesar 390 ohm.

Dengan pengkondisian sinyal seperti gambar 3.3, maka kondisi yang diharapkan akan tercapai, yaitu: saat PLC memberikan *logic "1"* diujung pengkondisi sinyal

berlogic “1” juga, sedangkan bila PLC *High Impedance* maka pengkondisi sinyal memberikan *logic* ”0”.

### 3.1.1.2. Desain *Buffer*



Gambar 3.4. Desain *Buffer*

Untuk menghindari terjadinya drop tegangan, maka hasil dari pengkondisi sinyal menggunakan *optocoupler*, dimasukkan ke *buffer* (Gambar 3.4). Selain itu metode ini juga berfungsi untuk menyesuaikan tegangan *output* dari *optocoupler* dengan karakteristik arus *input* dari DAC (baik *high* maupun *low level*). Hal ini terkait dengan *output current sink* dan *source* dari *op-amp* (LM324).

### 3.1.1.3. DAC (*Digital to Analog Converter*)

Pada pembuatan rangkaian DAC ini digunakan sebuah IC DAC 0808 yang bekerja pada system 8 bit yang mempunyai 16 pin dan sebuah IC LF 351 8 pin yang berfungsi untuk mengubah arus menjadi tegangan. Besarnya tegangan yang akan dihasilkan oleh DAC ( tegangan yang diinginkan sebesar 2V ) dapat dihitung dengan perumusan sebagai berikut :

$$\frac{V_{ref}}{R} = I \quad (3.1)$$

Jika :  $V_{Ref} = 2V$

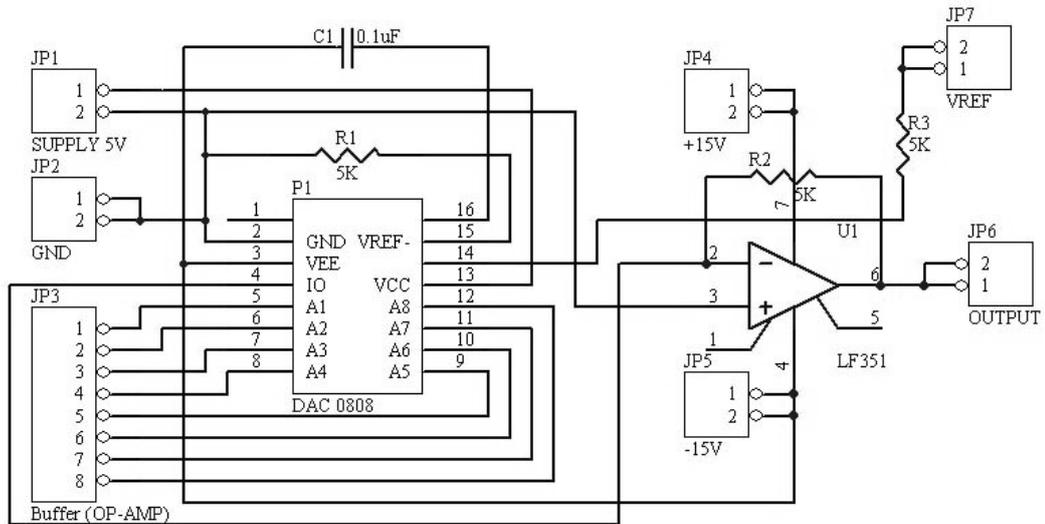
$$I = 2mA \text{ (datasheets)}$$

Maka :  $R = V_{ref} / I$

$$R = 2V / 2mA$$

$$R = 1 K\Omega$$

Berikut ini adalah gambar rangkaian dari DAC 0808 :

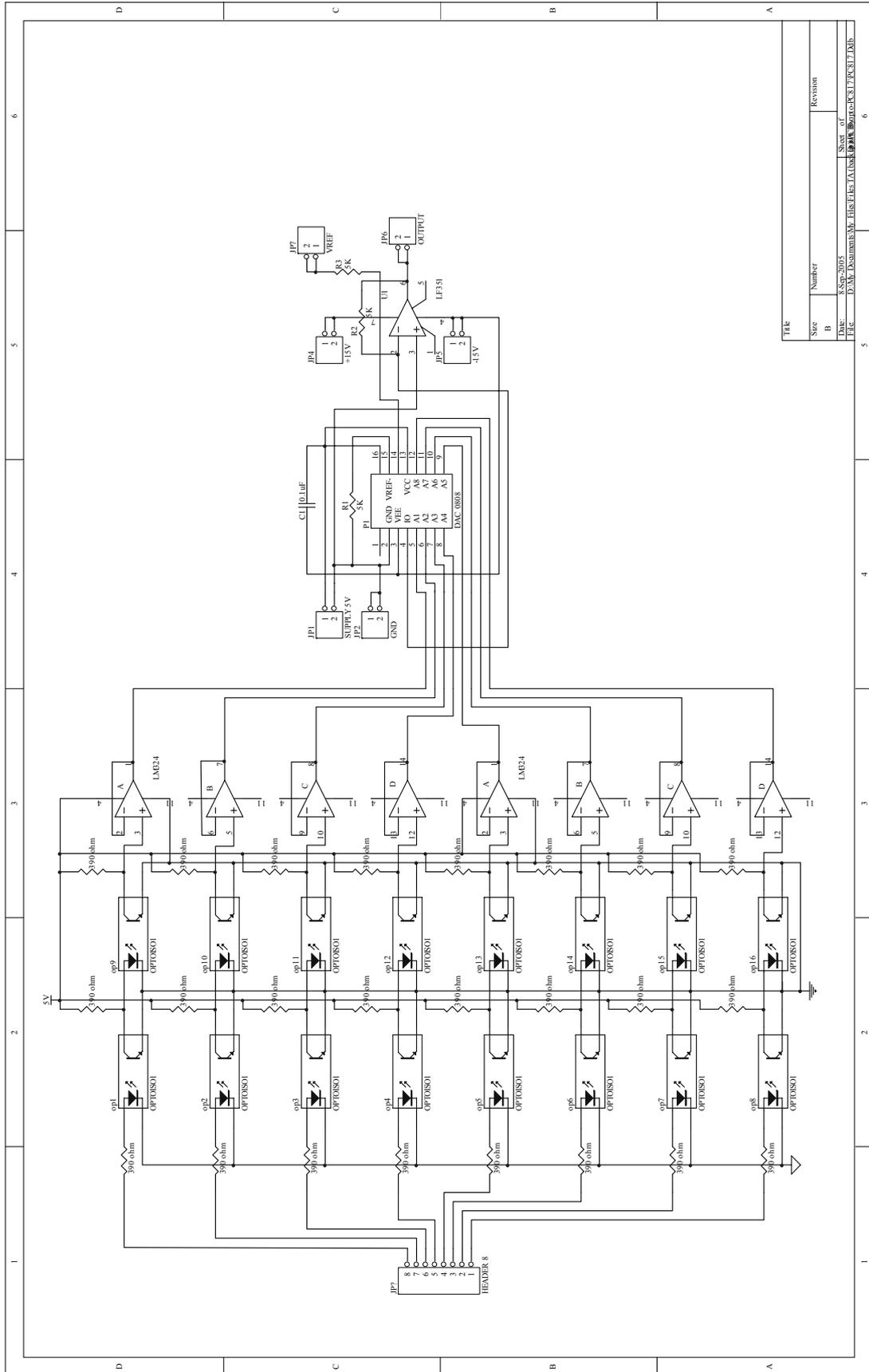


Gambar 3.5. Rangkaian DAC 0808

Sumber: *DAC0808 /DAC0807 /DAC0806 8-bit  $\mu$ P Compatible D/A Converters Datasheet*. California: National Semiconductor, Oktober 1989.

#### 3.1.1.4. Schematic Hardware Output

Berikut ini adalah desain *hardware output* sistem secara lengkap mulai dari output PLC sampai dengan DAC untuk dihubungkan dengan *Inverter Altivar* 16.



File	Size	Number	Revision
B			
Date:	8-Sep-2005		
File:	D:\M. Iqbal\m. Iqbal\KESIT\KESIT1.DXL		
		Sheet of	6

Gambar 3.6. Schematic Hardware Output

### 3.1.2. Inverter Altivar 16

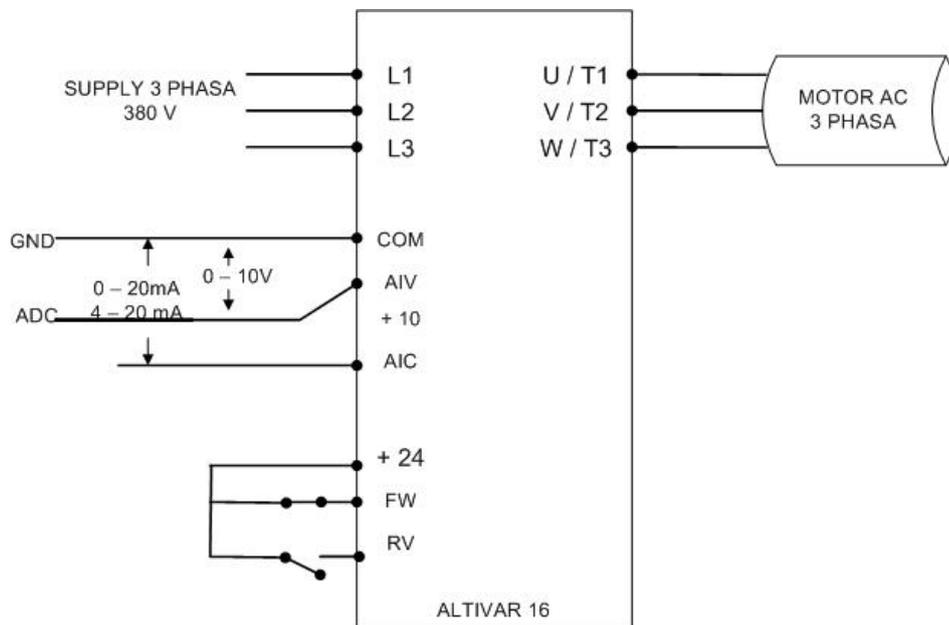
*Inverter Altivar 16* adalah suatu perangkat yang berfungsi sebagai alat untuk mengatur frekuensi dan kecepatan motor AC. Tegangan DC inputan yang digunakan untuk inputan *Inverter Altivar 16* tersebut sebesar 0 – 10V DC. Selain menggunakan tegangan DC, *Inverter Altivar 16* ini juga bisa menggunakan sumber arus untuk mengontrol kecepatan motor AC.

Untuk inputan *Inverter Altivar 16* digunakan supply AC 3 Phasa yang akan dihubungkan dengan L1, L2, dan L3. Outputan dari *Inverter Altivar 16* ini akan dihubungkan dengan motor AC 3 Phasa yang telah disusun secara *star* (380 V). Untuk mengontrol motor AC tersebut, diperlukan suatu inputan yang berupa arus atau tegangan DC.

Karena dalam pembuatan ini digunakan rangkaian DAC 0808, maka untuk mengontrol kecepatan motor AC tersebut digunakan inputan tegangan DC, tetapi inputan yang diberikan hanya sampai sebatas 2 volt. Hal ini disebabkan pada saat dilakukan pengujian terhadap motor AC yang telah dihubungkan dengan *Inverter Altivar*, dengan tegangan 2 Volt ini kecepatan motor AC yang dihasilkan sudah cukup cepat. Pada kecepatan inilah frekuensi yang dihasilkan oleh *encoder* sebesar 500 Hz.

Untuk mendapatkan arah putaran motor yang diinginkan, terdapat 2 buah *switch* yang dapat disetting sesuai dengan keinginan. Jika diinginkan arah putaran motor AC berputar searah dengan putaran arah jarum jam, maka +24V dihubungkan dengan *switch* FW. Sebaliknya jika diinginkan arah putaran motor AC tersebut berlawanan dengan arah putaran jarum jam, maka *switch* +24V dihubungkan dengan *switch* RV.

Berikut ini adalah gambar blok diagram dari *inverter altivar 16*:



Gambar 3.7. *Inverter Altivar 16*

### 3.1.3. Motor AC

Motor AC yang digunakan ini, dalam pemasangannya dirangkai secara *star* 380 Volt. Motor AC ini dapat berputar 1500 rpm pada kondisi tanpa beban.

Berikut ini adalah spesifikasi motor AC yang digunakan:

TECHO		3 - PHASE INDUCTION		MOTOR	
CODE : AE B			F # D 80		
4 POLE		1 HP 0.75 KW			
INFS	IP 54	AMB 40 C	50 HZ		
1500 RPM		STAR 380 V	DELTA 220 V		
BS 4999 & 5000		2.14 A	3.70 A		
CONT. RATING	1990				
BRG. NO . 62042Z					
SERI NO OC4KG40					

Gambar 3.8. *Nameplate Motor AC 3 Phasa TECHO*

#### 3.1.4. Encoder

Pada perencanaan *encoder* ini, digunakan sebuah piringan yang terbuat dari bahan *acrylic* yang bentuknya dibuat lingkaran dengan diameter 10 cm. Pada tepi piringan *encoder* tersebut, diberi lubang sebanyak 99 lubang. Hal ini dilakukan supaya sensor yang dipasang pada piringan tersebut dapat mendeteksi jumlah pulsa yang dihasilkan dalam setiap detik. Dari jumlah pulsa yang dihasilkan tiap detik tersebut, dapat diketahui besarnya frekuensi yang dihasilkan oleh putaran motor AC.

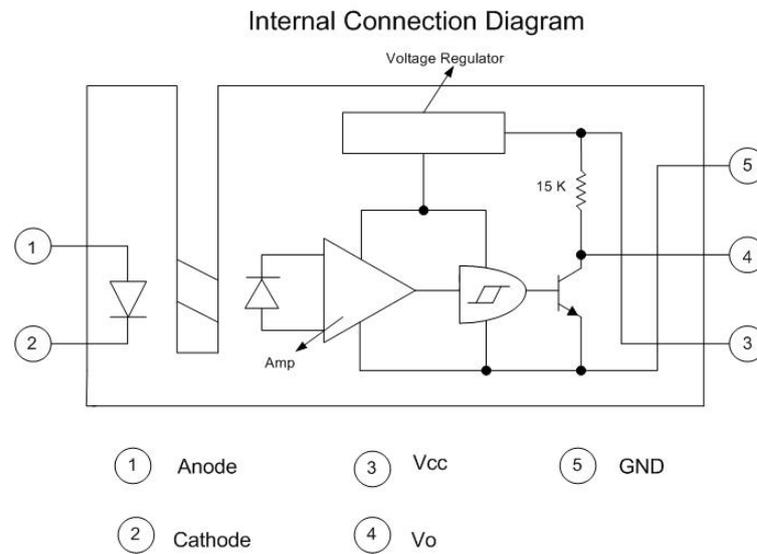


Gambar 3.9. *Acrylic*

Karena dalam pembuatan lubang pada *encoder* sebanyak 99 lubang, maka dapat diketahui bahwa dalam 1 kali putaran terdapat 100 gelombang, sedangkan pada inputan *inverter altivar 16* pada saat diberi tegangan 2 volt menghasilkan 500 gelombang per detik. Dari sini kita dapat memperoleh banyaknya gelombang tiap menit yaitu  $500 * 60 = 30.000$  gelombang per menit. Sehingga didapatkan bahwa kecepatan motor saat diberi tegangan 2 volt setara dengan 300 rpm ( $30.000/100$ ).

Sensor yang digunakan untuk mendeteksi pulsa pada *encoder* tersebut adalah *photointerrupter* yang memiliki jarak 3mm. Tegangan input yang digunakan untuk menjalankan *photointerrupter* ini sebesar 0 – 17 volt, tetapi pada pembuatan encoder kali ini, digunakan tegangan input sebesar 5 volt.

Berikut ini adalah gambar bagan dari *photointerrupter*:



Gambar 3.10. *Photointerrupter*

### 3.1.5. Desain *Hardware Input*

#### 3.1.5.1. F to V

Pada perencanaan alat ini digunakan rangkaian pengubah frekuensi menjadi tegangan atau lebih dikenal dengan istilah rangkaian F to V. Jenis IC yang digunakan adalah IC LM 331. IC LM 331 ini dapat mudah didapatkan dipasaran dengan harga yang tidak terlalu mahal. IC LM 331 ini mampu mengkonversi frekuensi sampai 10KHz (batas maksimum). Untuk mendapatkan tegangan yang diinginkan supaya alat tersebut dapat bekerja dengan baik, maka diperlukan perhitungan sebagai berikut dengan rumus sebagai berikut:

$$V_{out} = F_{in} \times 2.09 \times \frac{RL}{RS} \times (R_t.C_t) \quad (3.2)$$

$V_{out}$  yang diinginkan adalah sebesar 2 Volt, karena dengan tegangan 2 Volt ini kecepatan yang dihasilkan sudah cukup cepat. Pada rangkaian DAC 0808 ini digunakan resistor dan kapasitor sebagai berikut :

$F_{in} = 500 \text{ Hz}$  ( frekuensi yang akan dikonversi)

$RL = 100 \text{ K Ohm}$

$R_t = 6.8 \text{ K Ohm}$

$$C_t = 0.01 \text{ uF}$$

$$V_{out} = 2V \text{ ( yang diinginkan )}$$

Sesuai dengan perumusan 3.4, didapat perhitungan sebagai berikut:

$$2 V = 500 \times 2.09 \times \frac{100K}{RS} \times \{6.8 K \times (0.01 \times 10^{-6})\}$$

$$RS = \frac{500 \times 2.09 \times 100K \times (6.8K \times 10^{-8})}{2}$$

$$RS = \frac{500 \times 209 \times 6.8 \times 10^{-2}}{2}$$

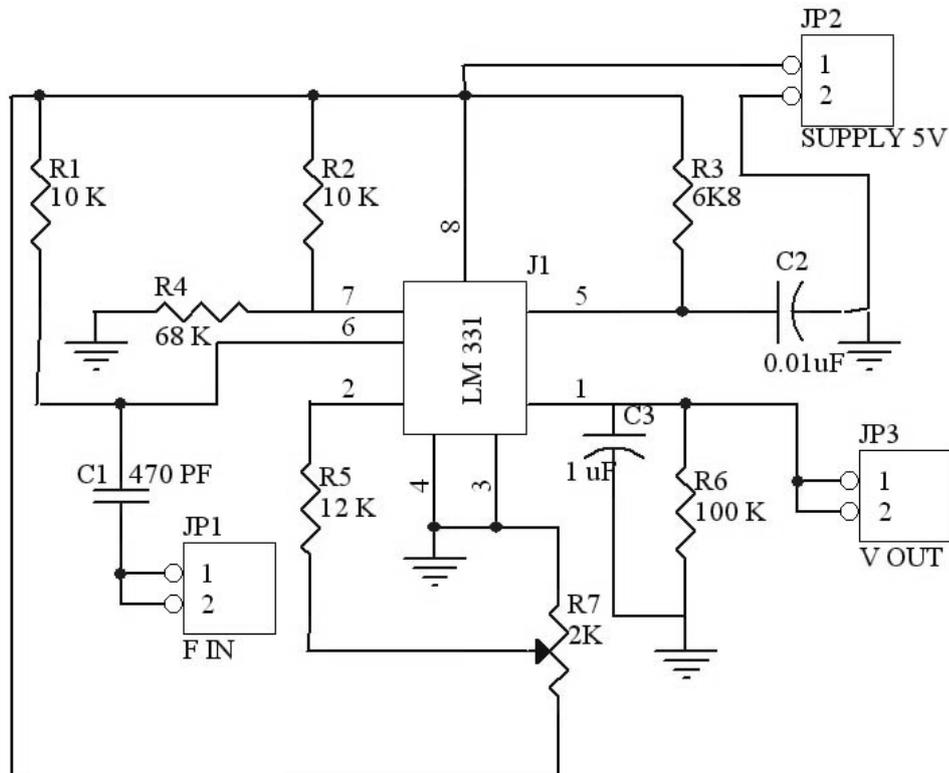
$$RS = 250 \times 209 \times 69 \times 10^{-3}$$

$$RS = 3605.25 \text{ Ohm}$$

$$RS = 3,6 \text{ K Ohm}$$

Pada pembuatan rangkaian F to V ini digunakan variabel resistor dengan nilai 5 K Ohm, sehingga dapat diatur sesuai dengan resistansi yang diinginkan. Pada pengujian rangkaian F to V ini sebaiknya tidak didekatkan dengan motor AC, hal ini disebabkan medan yang dihasilkan oleh motor AC dapat mempengaruhi frekuensi yang akan diuji.

Berikut ini adalah gambar rangkaian dari frekuensi ke tegangan :

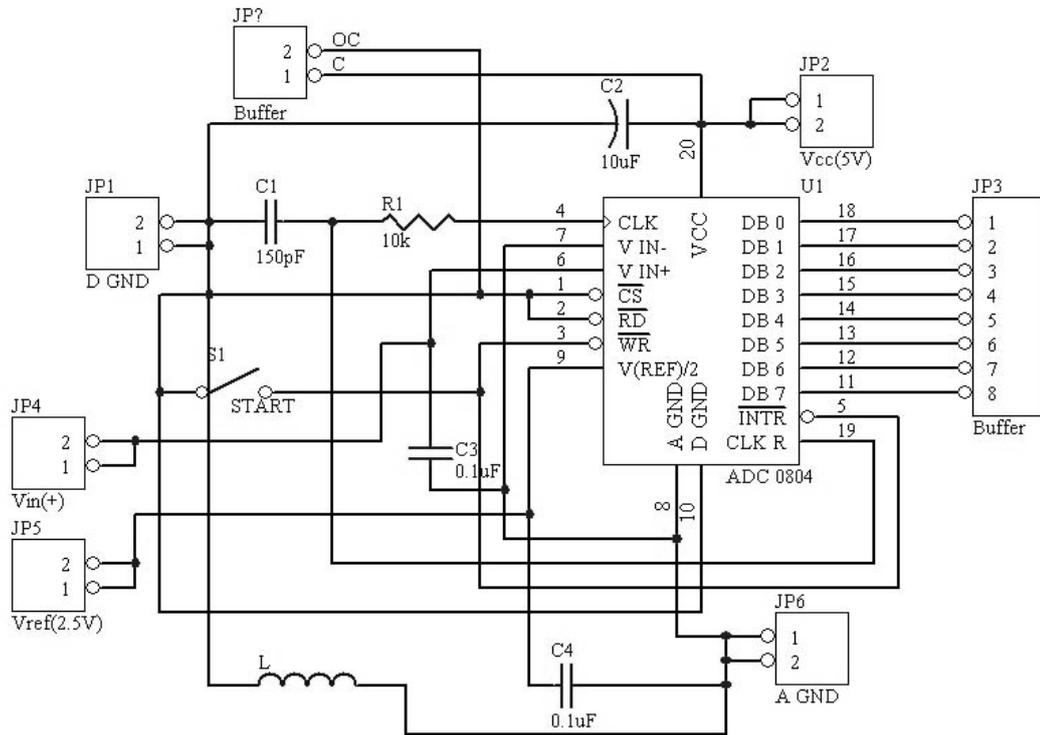


Gambar 3.11. Rangkaian F to V

Sumber: National Semiconductor. *LM 331 Datasheets*. 19 April 2006.  
<http://www.alldatasheet.com/datasheet-pdf/pdf/8641/NSC/LM331.html>.

### 3.1.5.2. Rangkaian ADC (*Analog-to-Digital Converter*)

PLC tidak dapat membaca tegangan *analog* yang dihasilkan oleh *output* LM331, sehingga tegangan *analog* itu harus diubah menjadi tegangan *digital* agar dapat dibaca oleh PLC. Untuk itu, diperlukan rangkaian ADC (*Analog-to-Digital Converter*) yang dapat mengubah tegangan *analog* menjadi tegangan *digital*. Rangkaian ADC ini menggunakan ADC0804 yang dapat menghasilkan 8 bit data biner (tegangan *digital*, 0V untuk logika '0' dan 5V untuk logika '1'), yang dirangkai seperti pada gambar 3.12.



Gambar 3.12. Rangkaian ADC0804

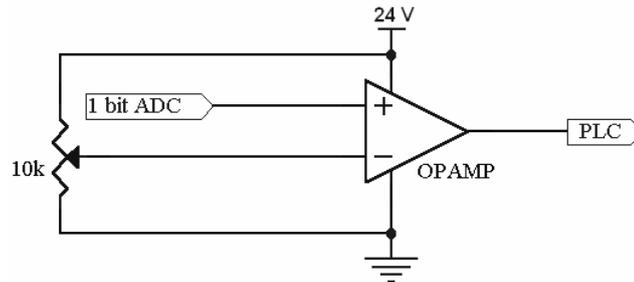
Sumber: *ADC0801 / ADC0802 / ADC0803 / ADC0804 / ADC0805 8-bit  $\mu$ P Compatible A/D Converters Datasheet*. California: National Semiconductor, Oktober1989.

### 3.1.5.3. Buffer

Permasalahan yang timbul adalah ketidakcocokan karakteristik antara ADC dan *op-amp* sebagai komponen dari *comparator*, di mana *input op-amp* memiliki impedansi yang tinggi, sehingga arus yang masuk ke *op-amp* rendah (orde *nanoampere*). Tetapi ADC memiliki arus yang agak tinggi yaitu dalam orde mikro untuk *logic "1"* dan orde mili untuk *logic "0"*. Untuk itu digunakan 74LS573 yang difungsikan sebagai *buffer*.

### 3.1.5.4. Penyelaras Tegangan (*Comparator*)

*Comparator* dipakai untuk menyesuaikan *level* tegangan antara ADC dan PLC. ADC bekerja pada *level* tegangan 5 VDC, sedangkan PLC bekerja pada *level* tegangan yang lebih tinggi yaitu 9 - 28 VDC.



Gambar 3.13. Desain *Comparator*

Dengan rangkaian tersebut bisa didapatkan *level* tegangan yang sesuai untuk PLC, di mana pada saat ADC memberikan *logic*, *comparator* akan membandingkan dengan  $V_-$ . Apabila  $V_-$  lebih kecil, maka *comparator* akan memberi *logic* "1" (sebesar VCC). Demikian sebaliknya bila  $V_-$  lebih besar, maka *comparator* memberikan *logic* "0" (0 V).

Pada pin  $V_-$  diberikan  $V_R = 10 \text{ kohm}$ , untuk membuat tegangan referensi yang digunakan sebagai unsur pembanding dari *output* ADC.

Contoh: *Output* ADC adalah 0 V (*logic* "0") dan 5 V (*logic* "1")

Dengan demikian tegangan referensi pada  $V_-$  harus ada pada *range* tegangan di antara kedua tegangan pada  $V_+$  (0 – 5 V). Karena jika tegangan tegangan pada  $V_-$  lebih besar dari 5 V, maka *comparator* akan selalu memberi *logic* "0".

#### 3.1.5.5. *Schematic Hardware Input*

Berikut ini adalah desain *input* dari awal proses pengolahan ADC sampai dengan terhubung ke PLC.

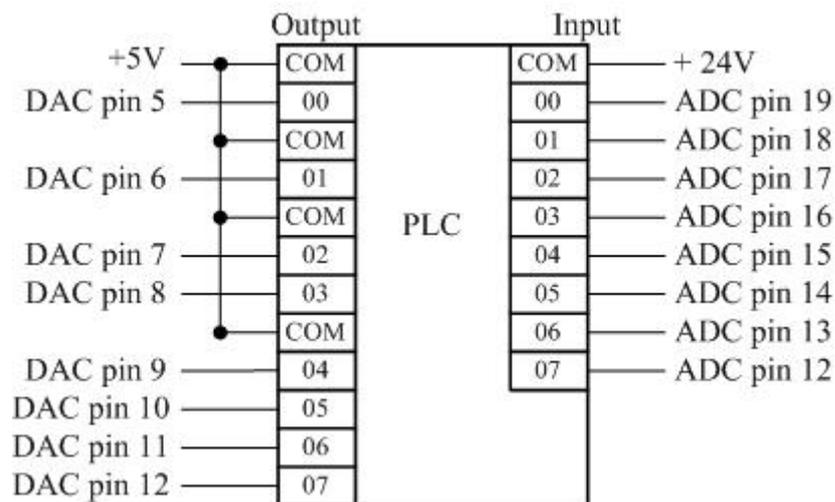


### 3.2. Perencanaan Software

Pada bagian ini akan dibahas dua desain yaitu desain *software* dengan PLC OMRON CPM1 lengkap dengan aplikasi SCADA sistemnya.

Untuk desain *software* pada PLC OMRON CPM1, *software* dibagi menjadi dua bagian yaitu program aplikasi yang dijalankan oleh PLC dan program SCADA yang dijalankan oleh komputer dengan dukungan SCADA *software*.

Berikut ini adalah gambar *wiring* PLC :



Gambar 3.15. *Wiring* PLC

#### 3.2.1. Program aplikasi

Sebelum memulai untuk membahas desain *software* opada PLC, perlu diketahui terlebih dahulu *input/output list* yang terhubung ke PLC.

##### 1. *Input*

- ADC yang dialamatkan pada IR00000 – IR00007
- *Start* yang dialamatkan pada IR20001
- *Stop* yang dialamatkan pada IR20002

##### 2. *Output*

- DAC yang dialamatkan pada IR010

##### 3. Memori

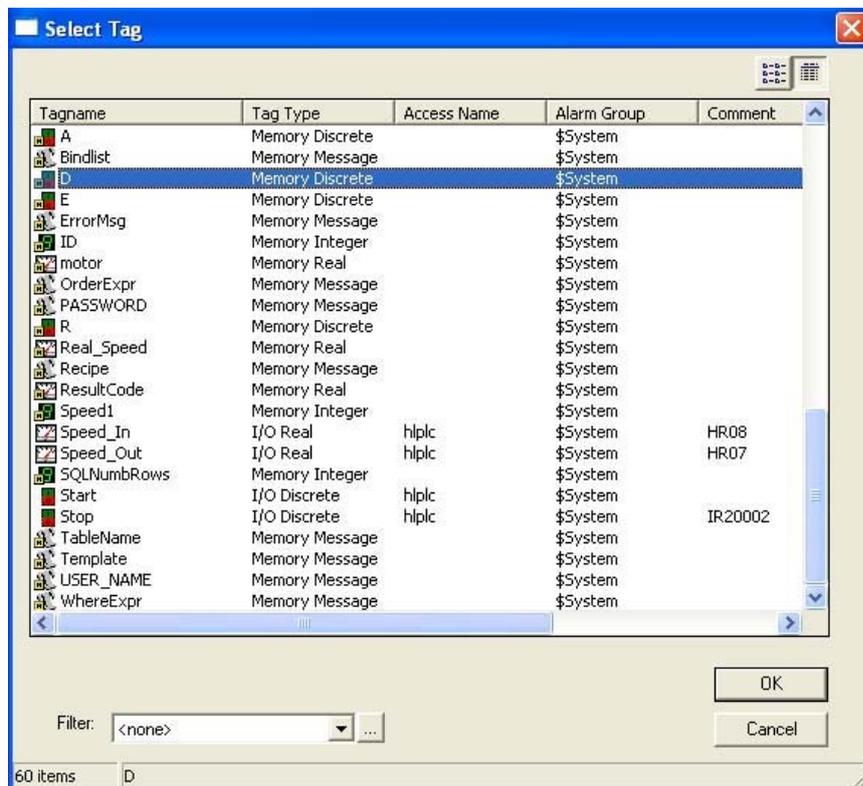
- *Input* IR000 dialamatkan pada DM0502
- *Output* IR010 dialamatkan pada DM0503

Tujuan dari program aplikasi tersebut adalah untuk memasukkan nilai setpoint dan mengambil nilai dari sensor. Program dibuat dengan bantuan *software* SYSMAC 3,0 yang menggunakan bahasa *ladder logic*. *Ladder* akan dijelaskan dalam lampiran.

### 3.2.2. Desain SCADA Sistem

Untuk melakukan fungsi kontrol dan monitor pada *plant*, digunakan SCADA Sistem dengan bantuan SCADA *software* yaitu *Wonderware InTouch* dan *I/O Server*.

Sebelum memulai lebih lanjut, ada baiknya diketahui terlebih dahulu *tagname* yang digunakan dalam pembuatan SCADA tersebut.



Gambar 3.16. *Tagname*

*Tagname-tagname* tersebut digunakan sebagai media yang mendukung untuk mengambil dan mengirim data dari dan ke PLC, membuat animasi grafis untuk menggambarkan proses yang terjadi, perhitungan dan media kontrol untuk mengontrol PLC. Untuk lebih jelasnya kegunaan masing-masing akan dibahas pada tiap *window*.

Desain SCADA tersebut terdiri dari beberapa *window* yang memiliki fungsi masing-masing:

1. *Login*



Gambar 3.17. *Login*

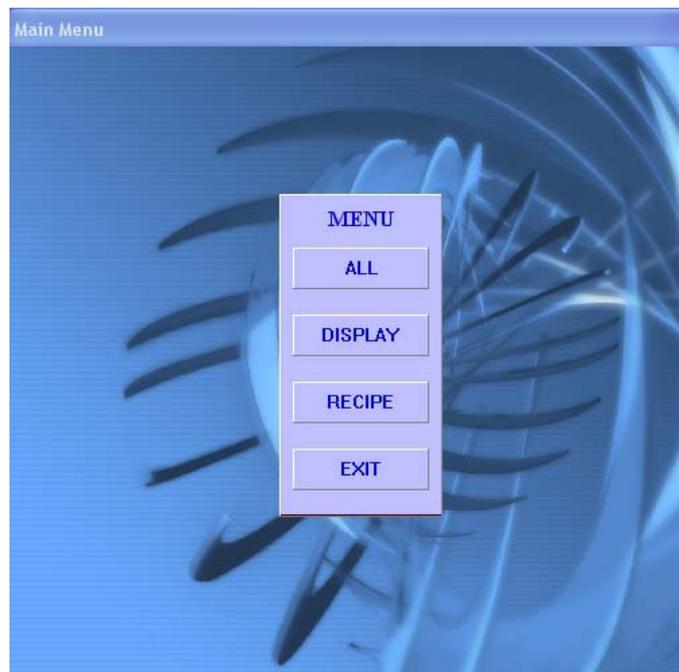
*Window* tersebut adalah tampilan utama sebagai gerbang untuk masuk ke sistem. Pada bagian tersebut terdapat “*username*” dan “*password*” yang dibutuhkan untuk membuka *window-window* berikutnya.

Ada 2 “*username*” dan “*password*” yang digunakan yaitu “operator” dan “petra” serta “administrator” dan “elektro”. Kedua kombinasi “*username*” dan “*password*” tersebut memiliki kegunaan yang hampir sama, namun yang membedakan adalah kombinasi “operator” dan “petra” hanya memiliki akses untuk menampilkan *window* “Display”. Sedangkan untuk kombinasi “administrator” dan “elektro” memiliki akses ke semua *window*.

Karena digunakan sebagai gerbang utama yang bisa mengenali “*username*” dan “*password*” maka pada *script*-nya diketikkan beberapa perintah yang bisa mengenali “*username*” dan “*password*” serta untuk menutup *window* yang lain.

```
IF USER_NAME == "operator" AND PASSWORD == "petra" THEN
  Show "Main Menu";
  Hide "Login";
ELSE
  IF USER_NAME == "administrator" AND PASSWORD == "elektro" THEN
    Show "Main Menu";
    Hide "Login";
  ENDIF;
ENDIF;
```

## 2. Menu



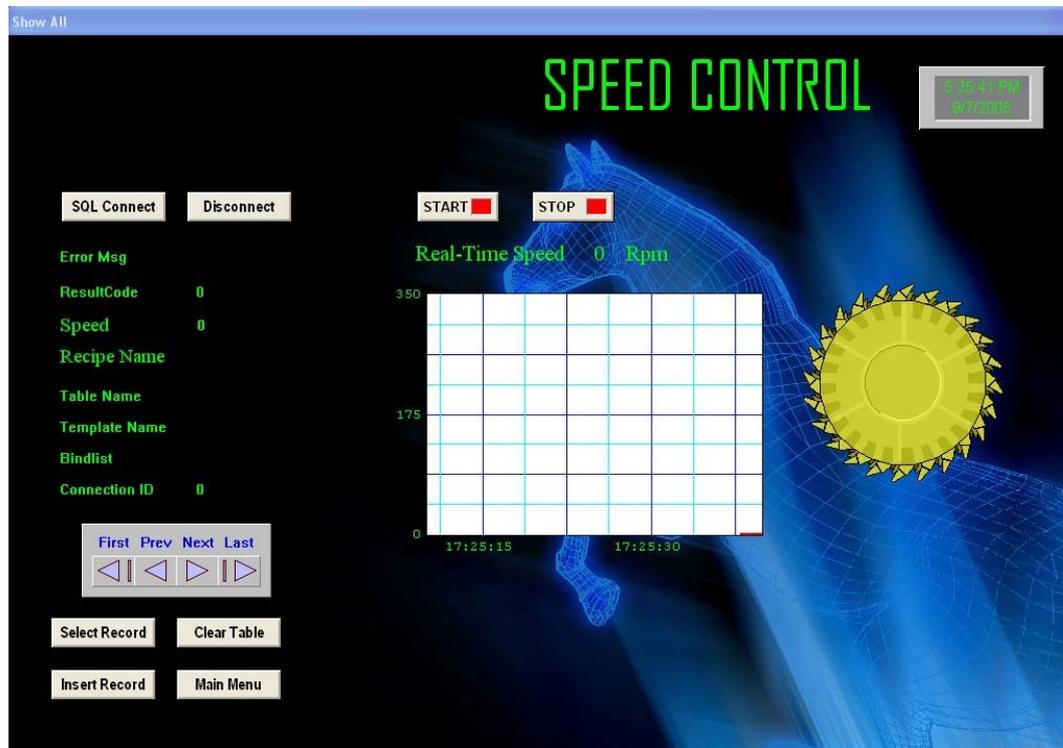
Gambar 3.18. Menu

*Window* menu memiliki beberapa bagian yang menghubungkan dengan *window* yang lain seperti yang tertera pada tombolnya. Hal tersebut dilakukan dengan meng-set *touch\_pushbutton* pada masing-masing *push button* dan memilih *window* mana yang ingin ditampilkan.

Karena menu *window* adalah yang pertama dibuka setelah *login* maka pada bagian *script*-nya diketikkan program untuk clear “*username*” dan “*password*”.

```
IF A == 1 THEN
    Show "Show All";
    USER_NAME = " ";
    PASSWORD = " ";
ENDIF;
IF D == 1 THEN
    Show "Display";
    USER_NAME = " ";
    PASSWORD = " ";
ENDIF;
IF R == 1 THEN
    Show "Recipe";
    USER_NAME = " ";
    PASSWORD = " ";
ENDIF;
IF E == 1 THEN
    Show "Login";
    USER_NAME = " ";
    PASSWORD = " ";
ENDIF;
```

### 3. *Show All*



Gambar 3.19. *Show All*

Pada window “*Show All*” ditampilkan semua animasi yang digunakan, termasuk animasi, nilai *setpoint*, nilai kecepatan, *recipe*, koneksi *database* dan *trend*. Pada bagian *trend* menampilkan data dari *set\_point* dan *real\_time speed* dari motor. Dalam koneksi *database* diperlukan *script*, *script* tersebut akan dimasukkan ke dalam *Touch Pushbuttons* “*Action*” :

1. Pada *SQL Connect* (digunakan untuk *connect* pada *Microsoft SQL Server*)  
 ResultCode = SQLConnect( ID,"DSN=Connect InTouch; DB=Runtime;  
 UID=sa; PWD=wit" );  
 TableName = "Table1";  
 Template = "Template1";  
 Bindlist = "List1";
2. Pada *Disconnect* (digunakan untuk *disconnect* pada *Microsoft SQL Server*)  
 ResultCode = SQLDisconnect( ID );  
 ID = 0;  
 ResultCode = 0;

3. Pada *Select Record* (digunakan untuk memilih kolom pada tabel di dalam *Microsoft SQL Server*)
 

```

      ResultCode = SQLSelect( ID,TableName,Bindlist,WhereExpr,OrderExpr);
      SQLFirst(ID);
      SQLNumbRows = SQLNumRows(ID);
      
```
4. Pada *Clear Table* (digunakan untuk menghapus semua data pada tabel di dalam *Microsoft SQL Server*)
 

```

      ResultCode = SQLClearTable( ID,TableName) ;
      Speed1 = 0;
      Recipe = " ";
      
```
5. Pada *Insert Record* (digunakan untuk menyimpan data pada tabel di dalam *Microsoft SQL Server*)
 

```

      ResultCode = SQLInsert( ID,TableName,Bindlist) ;
      SQLNumbRows = SQLNumRows( ID );
      
```
6. Pada *First* (digunakan untuk memilih resep yang pertama)
 

```

      ResultCode = SQLFirst( ID) ;
      
```
7. Pada *Prev* (digunakan untuk memilih resep yang sebelumnya)
 

```

      ResultCode = SQLPrev( ID) ;
      
```
8. Pada *Next* (digunakan untuk memilih resep yang selanjutnya)
 

```

      ResultCode = SQLNext( ID) ;
      
```
9. Pada *Last* (digunakan untuk memilih resep yang terakhir)
 

```

      ResultCode = SQLLast( ID) ;
      
```

Untuk koneksi juga diperlukan pendefinisian tabel yang akan dibuat. Pendefinisian itu dibuat dalam *SQL Access Manager* pada *Application Explorer*. Pada *Bindlist* mendefinisikan *tagname* menjadi nama kolom, pada *Table Template* mendefinisikan tipe kolom.

Berikut ini adalah gambar *Bindlist* yang digunakan :

**Bind List Configuration**

Add Item Delete Item Modify Item Cancel OK

Bind List Name: List1 Save

Tagname.FieldName Column Name  
Recipe RecipeName

Tagname FieldName Move Up Move Down

Tagname.FieldName	Column Name
Recipe	RecipeName
Speed1	Speed

Gambar 3.20. *Bindlist*

Berikut ini adalah gambar *Template* yang digunakan :

**Table Template Configuration**

Add Item Delete Item Modify Item Cancel OK

Table Template Name: Recipe

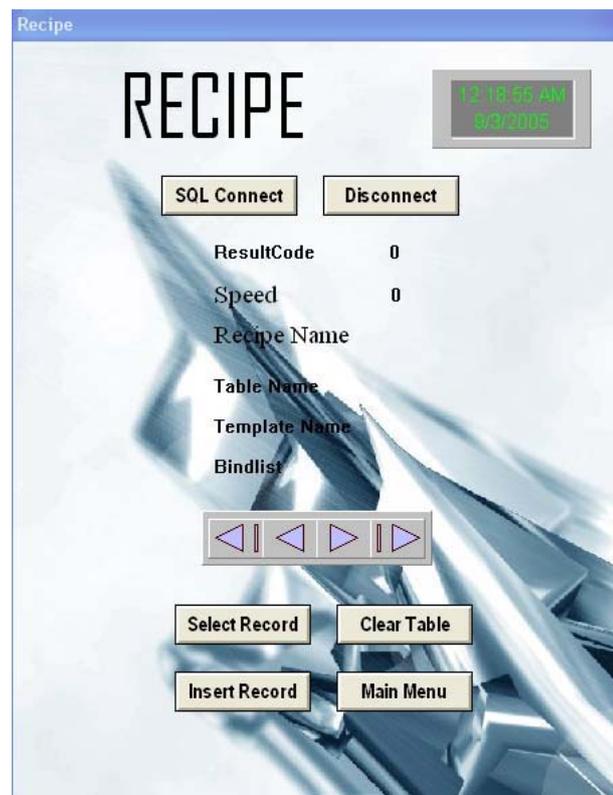
Column Name	Column Type	Length
NamaRecipe	Char	10

Index Type:  Unique  Non-Unique  None  Allow Null Entry

Column Name	Column Type	Length	Allow Null
NamaRecipe	Char	10	Null
Speed1	Numeric	10	Null

Gambar 3.21. *Template*

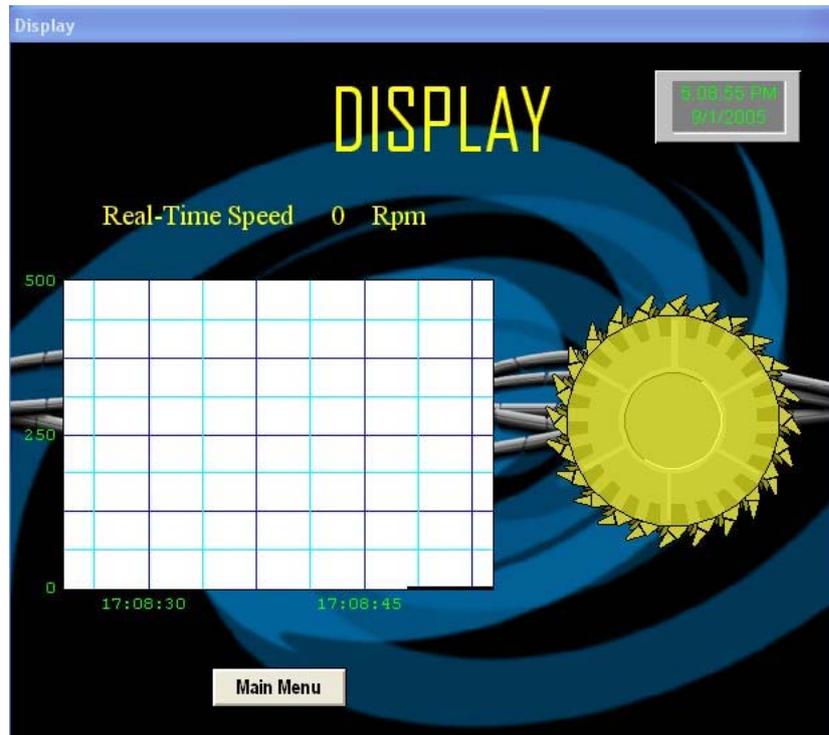
#### 4. *Recipe*



Gambar 3.22. *Recipe*

Window “Recipe” merupakan bagian dari window “Show All” dan hanya berisi recipe dan koneksi database.

## 5. Display



Gambar 3.23. *Display*

Window "Display" merupakan bagian dari window "Show All" dan hanya berisi animasi dan trend. Pada bagian ini yang lebih ditonjolkan adalah grafik respon yang diberikan *real\_time trend* dan gerak dari motor. Pada grafik tersebut menampilkan data dari *set\_point* dan *real\_time speed*.