

BAB III

PERANCANGAN SISTEM

3.1. Perancangan Sistem.



Gambar 3.1 Blok Diagram Sistem

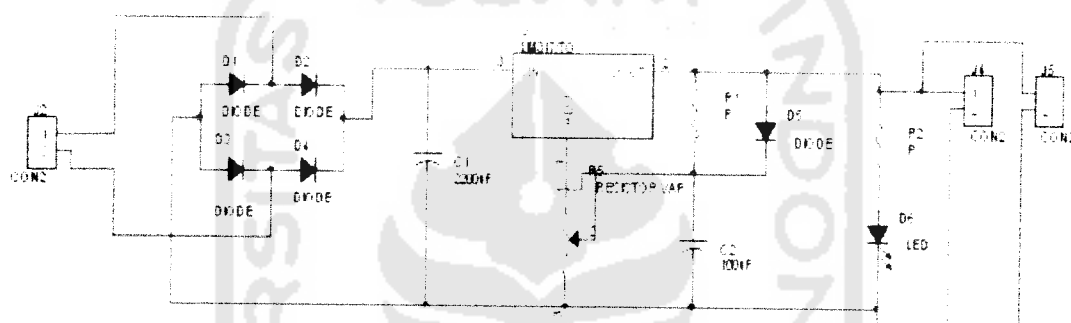
Dalam sistem ini PLC mengatur tegangan Motor DC dengan menggunakan metode PWM. Pertama-tama sinyal PWM (gelombang kotak) diatur dengan memasukan nilai *off duty cycle* pada instruksi PWM pada PLC. Output dari PLC yang berupa PWM tersebut akan menjadi input dari driver Motor DC untuk mengendalikan tegangan Motor DC.

Kecepatan yang dihasilkan oleh Motor DC dibaca dengan menggunakan sensor *optocoupler*. Sensor tersebut digunakan sebagai inputan PLC untuk menghitung kecepatan Motor DC. Proses penghitungan kecepatan Motor DC dilakukan dengan menggunakan instruksi *high speed counter* yang akan membaca pulsa yang dihasilkan sensor *optocoupler*. Pulsa tersebut kemudian dihitung dengan

mengatur parameter – parameter yang ada dalam program PLC sehingga dihasilkan kecepatan yang sesuai dengan *duty cycle*.

3.2 Perancangan *Hardware*

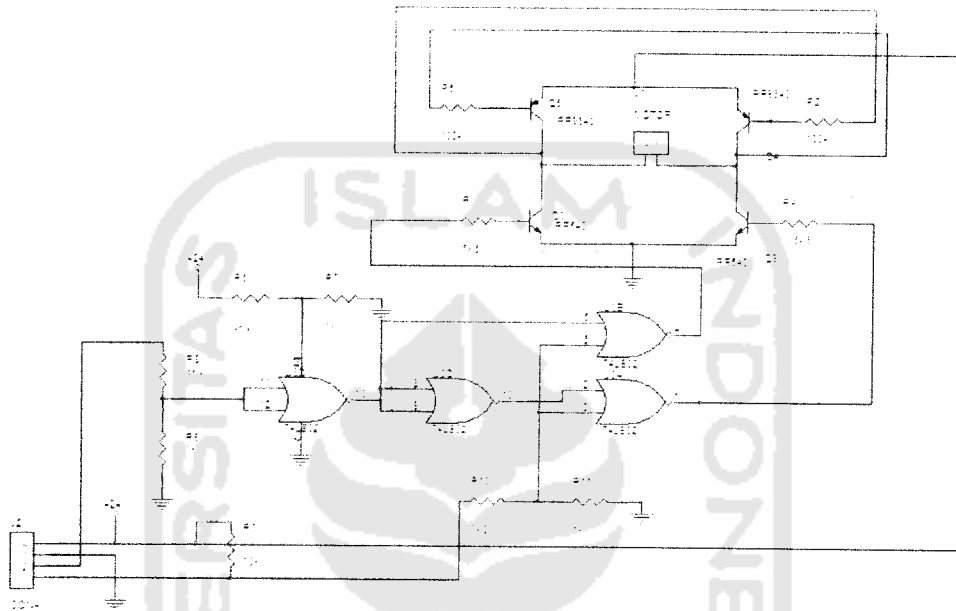
3.2.1. Rangkaian Power Supply



Gambar 3.2 Rangkaian Catu Daya

Power supply digunakan untuk menyuplai tegangan yang dibutuhkan oleh input/output PLC, Motor DC dan sensor *optocoupler*. Oleh karena itu untuk menghasilkan tegangan 24V digunakan IC *regulator* LM317 yang dapat menghasilkan tegangan 24V.

3.2.2. Rangkaian Driver Motor DC



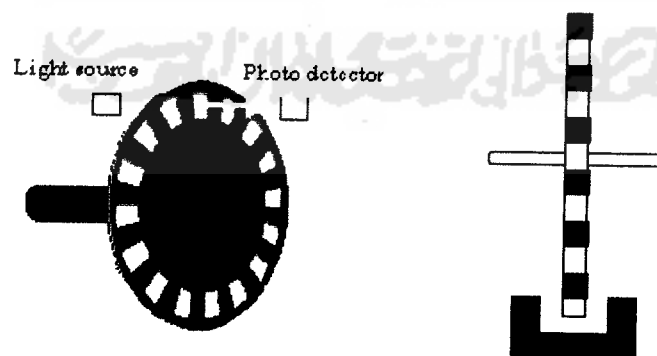
Gambar 3.3 Rangkaian Driver Motor Dc

Pada rangkaian driver motor diatas menggunakan rangkaian *half bridge* sehingga motor bisa berputar 2 arah kiri dan kanan secara bergantian. Pada rangkaian *half bridge*, hal yang tidak boleh terjadi adalah keempat buah transistor yaitu NPN kiri, NPN kanan, PNP kiri dan PNP kanan aktif secara bersamaan karena hal ini akan menghubungkan singkatkan sumber daya positif dan negatif, yang dapat mengakibatkan motor tidak dapat berputar. Untuk mencegah kondisi ini, rangkaian gerbang logika yang dibentuk oleh IC 74LS02 diatur sehingga NPN kiri dan NPN kanan aktif bergantian. Hal ini ditentukan oleh kondisi logika pada kaki 2 J2 sebagai penentu arah motor. Sedangkan untuk mengatur apakah motor dalam keadaan aktif atau tidak

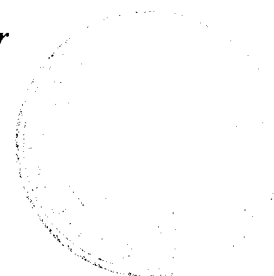
ditentukan oleh kaki 1 J2. Jika kondisi pada kaki 2 J2 berlogika 0, maka keluaran pada U3B akan berlogika 0 pula selama kaki 1 J2 aktif (berlogika 0). Hal ini akan mengakibatkan transistor NPN kiri non aktif sedangkan keluaran U3C akan berlogika 1 yang mengakibatkan keluaran U3A juga berlogika 1 dan transistor NPN kanan aktif. Kaki 1 J2 berfungsi untuk mengatur kecepatan motor dengan membangkitkan PWM yang dikeluarkan oleh output PLC.

3.2.3 Rangkaian *Optocoupler*

Untuk mengetahui atau mengukur kecepatan putar motor digunakan sensor optik berupa sepasang pemancar dan penerima infra merah yang sering disebut *optocoupler*. Sensor optik ini membaca piringan berlubang yang dipasangkan dengan dikopel pada poros motor. Banyaknya lubang sangat mempengaruhi ketelitian pembacaan kecepatan. Lubang yang dibuat sebanyak 12 buah. Ini dikarenakan keterbatasan alat dan bahan, dan juga untuk memudahkan dalam pemrograman. Pemasangan sensor ditunjukkan pada gambar berikut:



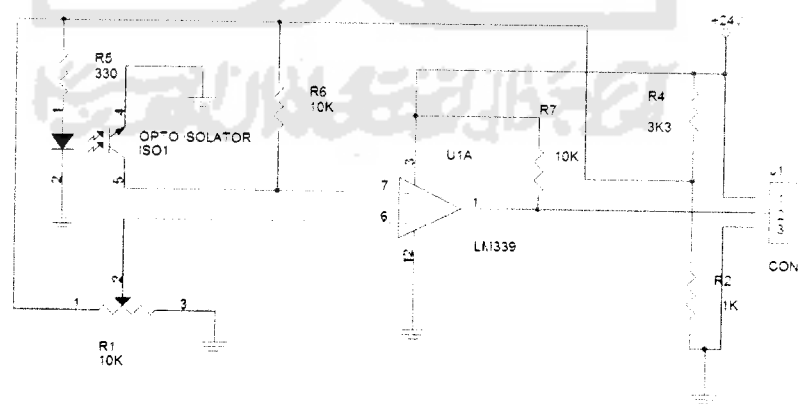
Gambar 3.4 Piringan *Encoder*



Optocoupler terdiri atas led inframerah dan fototransistor. Led inframerah yang digunakan hanya mampu melewati arus maksimal sebesar 20 mA. Oleh karena itu perlu ditambahkan sebuah resistor sebagai pembatas arus. Besaran nilai resistor minimal ditentukan dengan menggunakan perhitungan sebagai berikut:

$$R = \frac{V}{I} = \frac{24}{20mA} = 1,2k\Omega \quad (3.1)$$

Dalam perancangan digunakan resistor dengan resistansi 1 k Ω . Dengan nilai resistansi ini, arus yang masuk ke LED inframerah sebesar 15,15 mA. Pada bagian fototransistor, kaki kolektor diberikan resistor *pull-up* 10 k Ω dan dihubungkan ke kaki komparator menggunakan IC LM339. Pemilihan IC ini dilakukan dengan pertimbangan yaitu IC LM339 terdiri dari 4 buah komparator yang masing-masing berdiri sendiri-sendiri.

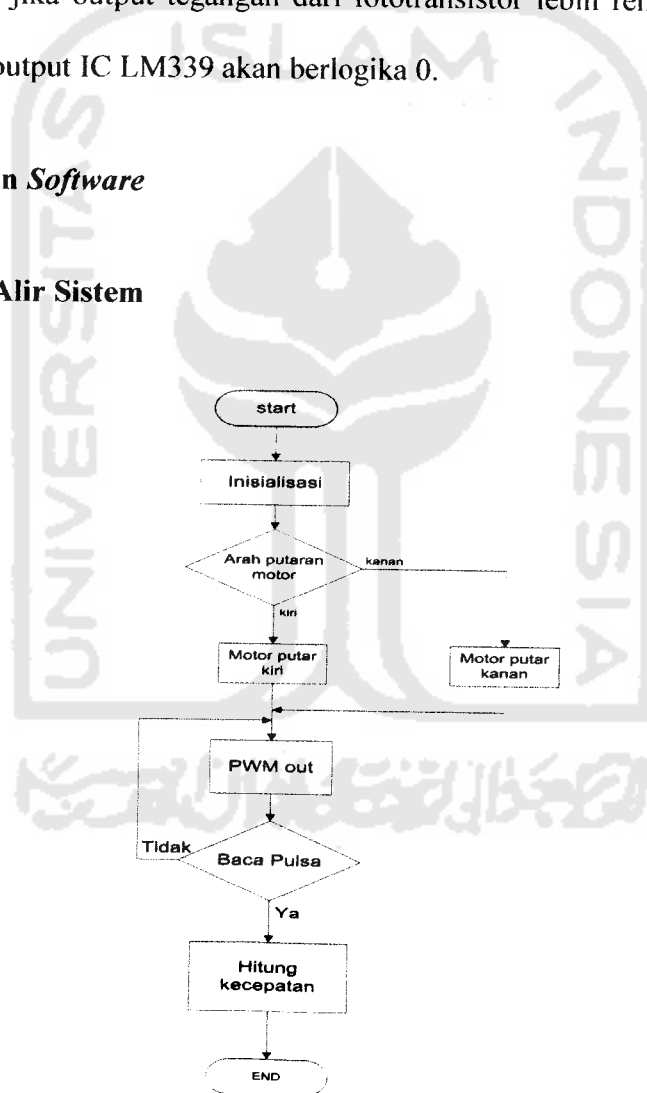


Gambar 3.5 Rangkaian *Optocoupler*

Rangkaian ini berfungsi sebagai pendeteksi apakah ada pengukuran yang terjadi atau tidak. Dalam hal ini sebagai indikator dipakai tegangan output dari rangkaian fototransistor. Apabila tegangan output dari rangkain fototransistor ini lebih besar dari tegangan referensi, maka output IC LM339 akan berlogika 1 begitu juga sebaliknya jika output tegangan dari fototransistor lebih rendah dari tegangan referensi maka output IC LM339 akan berlogika 0.

3.3 Perancangan *Software*

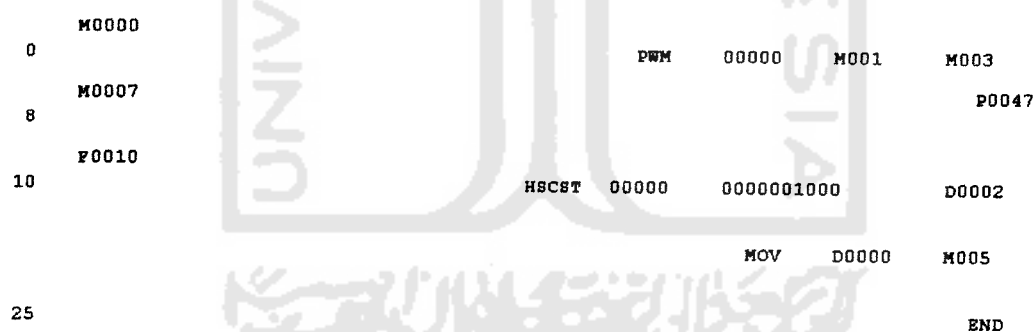
3.3.1 Diagram Alir Sistem



Gambar 3.6 Diagram Alir Sistem

3.3.2 Diagram Ladder

Untuk membuat suatu program PLC harus menggunakan *software* yang mendukung PLC tersebut. *Software* yang mendukung PLC MASTER-K120S adalah KGL WIN365. Sebelum program dirancang penggunaan input dan output pada PLC harus di deskripsikan terlebih dahulu. Pendeskripsian ini sangat penting karena pada perancangan *software* nanti kita harus mengetahui alamat input ataupun alamat output mana yang digunakan. Pada perancangan ini membutuhkan 1 alamat input P00, 2 alamat output P40 dan P47, dan penggunaan 5 memori sebagai saklar dan juga sebagai tempat menyimpan data yaitu M000, M001, M003, M005, dan M007. Berikut adalah *ladder* diagram perancangan pengaturan kecepatan motor dc.



Gambar 3.7 Diagram *Ladder* Pengaturan Kecepatan Motor DC

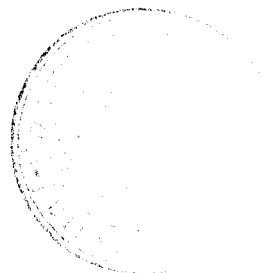
M000 dan M007 digunakan sebagai saklar untuk mengaktifkan sistem jika berlogika 1. M000 akan mengaktifkan instruksi PWM dan mengeluarkan sinyal PWM pada alamat output P40. Pada instruksi PWM alamat P40 merupakan Ch.0

yang digunakan untuk mengeluarkan sinyal PWM tersebut yang kemudian dihubungkan pada driver Motor DC. M001 digunakan untuk memasukan nilai periode untuk menentukan berapa frekuensi yang akan digunakan untuk mengendalikan kecepatan motor dc. Pada M001 dapat kita isi dengan periode 50ms sehingga dihasilkan frekuensi 20Hz.

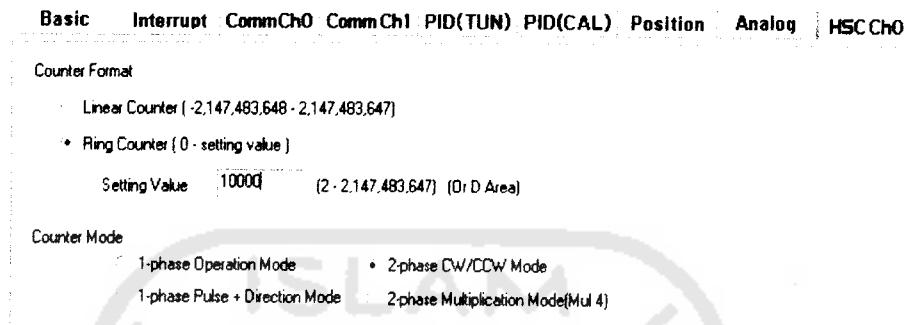
$$f = \frac{1}{T} = \frac{1}{50ms} = 20Hz \quad (3.2)$$

M003 digunakan untuk menentukan berapa persen duty cycle yang digunakan untuk mengatur lebar pulsa PWM. Dalam menentukan arah putaran motor saklar M007 diberi logika 1 sehingga koil P47 yang terhubung dengan driver Motor DC aktif dan mengakibatkan motor putar arah.

Untuk pembacaan pulsa yang dihasilkan oleh rangkaian *optocoupler* menggunakan instruksi HSCST yang diaktifkan oleh alamat F10. Alamat F10 berfungsi agar instruksi HSCST langsung aktif (*always on*). Output dari *optocoupler* dihubungkan ke alamat input P000. Alamat input P000 merupakan Ch.0 yang digunakan sebagai parameter input untuk instruksi HSCST. Pulsa yang terbaca pada Ch.0 kemudian akan dibaca dan dihitung sehingga menghasilkan nilai Rpm yang sesuai dengan pulsa masukan. Nilai Rpm ini dimasukan di alamat D0. Proses penghitungan Rpm dilakukan dengan melakukan setingan pada parameter-parameter HSCST yaitu :



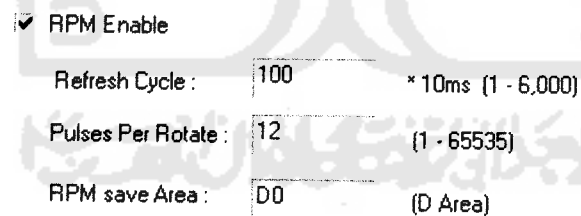
1. Setting format counter dan mode counter



Gambar 3.8 Setting Format Counter dan Mode Counter

Pada format counter dipilih ring counter untuk membatasi pulsa yang terbaca, pada settingan ini diisi 10000 pulsa. Sedangkan untuk settingan mode counter dipilih 2phase CW/CCW mode. Mode ini dipilih untuk motor 2 arah.

2. Setting Rpm



Gambar 3.9 Setting Rpm

Pada settingan Rpm *refresh cycle* diisi dengan nilai 100x10ms atau sama dengan 1 detik, artinya proses penghitungan pulsa dihitung dalam 1 detik. Jumlah

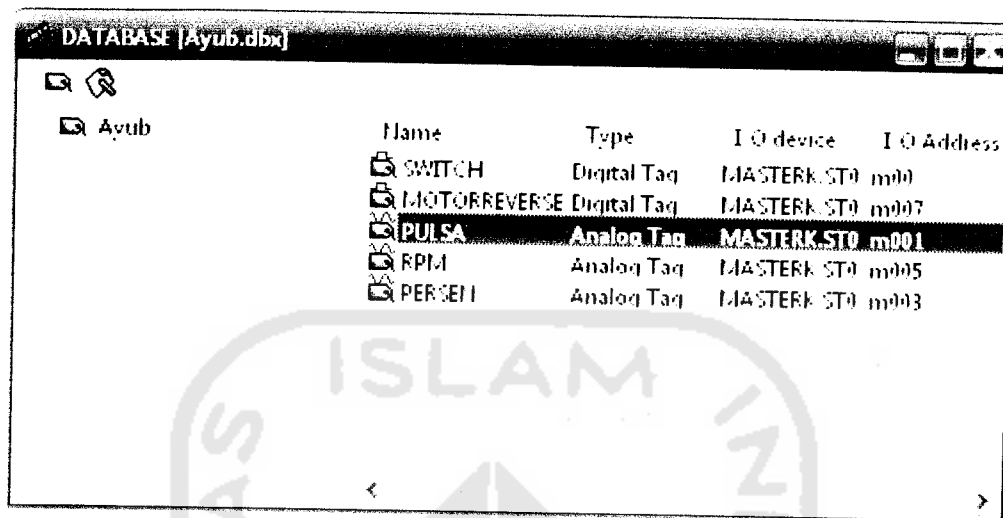
pulsa dalam 1 putaran diisi sesuai dengan lubang yang dibuat pada *encoder* yaitu 12 pulsa. Untuk menghitung Rpm dapat dihitung dengan menggunakan rumus :

$$Rpm = \frac{(Current\ Value - Last\ Value) \times 60.000}{pulse\ per\ rotate \times refreshcycle\ (ms)} \quad (3.3)$$

Nilai Rpm yang telah terhitung dan tersimpan di alamat D0 akan dipindahkan ke alamat M005 dengan menggunakan instruksi MOV. Hal ini dikarenakan agar nilai kecepatan yang telah dihitung dapat ditampilkan pada simulasi monitoring kecepatan Motor DC dengan menggunakan program CimonD dengan memanggil alamat M005.

3.3.3 Perancangan Simulasi

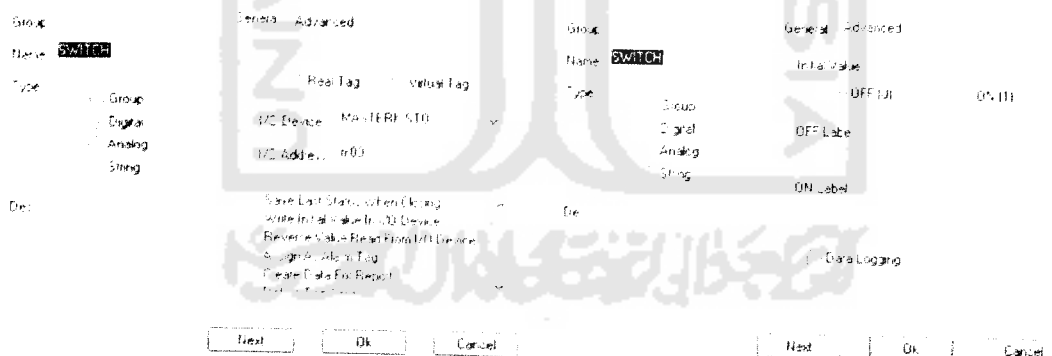
Untuk membuat simulasi pada perancangan ini diperlukan software yang mendukung untuk membuat simulasi secara real atau sesuai dengan keadaan suatu sistem. CimonD merupakan *software* simulasi yang mendukung PLC LG MASTER K120S dengan tool yang mudah dipahami. Proses pembuatan, pertama-tama dilakukan dengan membuat database pada input dan output dari diagram ladder PLC tersebut.



Name	Type	I/O device	I/O Address
SWITCH	Digital Tag	MASTERK-ST0	m00
MOTORREVERSE	Digital Tag	MASTERK-ST0	m007
PULSA	Analog Tag	MASTERK-ST0	m001
RPM	Analog Tag	MASTERK-ST0	m005
PERSEN	Analog Tag	MASTERK-ST0	m003

Gambar 3.10 Database Pada CimonD

Database pertama yang dibuat yaitu dengan nama SWITCH. Konfigurasi pada database ini adalah sebagai berikut :



Group: SWITCH

Type: Digital

I/O Device: MASTERK-ST0

I/O Address: m00

Real Tag: Value Tag:

OFF Label: 05.113

ON Label:

Buttons: Next, Ok, Cancel

Gambar 3.11 Konfigurasi Pada Database *Swicth*

Pada konfigurasi database *switch* diatas tipe yang dipilih adalah digital yang berarti On pada masukan logika 1 dan Off pada masukan 0. Pada bagian settingan general dipilih real tag agar sesuai dengan keadaan sistem. Untuk kolom bagian I/O Device

dipilih tipe PLC yang digunakan sedangkan untuk *I/O Address* diisi alamat memori pada inputan PLC yaitu alamat M00. Penggunaan alamat ini adalah untuk mengaktifkan alamat inputan pada PLC yang akan mengaktifkan instuksi PWM. Pada bagian settingan *advanced* untuk kolom *initial value* dipilih Off yang berarti Off pada kondisi awal.

Database kedua yang dibuat yaitu dengan nama MOTORREVERSE, konfigurasi data base ini hampir sama dengan database *switch*, hanya saja pengalamatannya yang berbeda. Pada database ini alamat memori yang dipanggil pada PLC yaitu M007 yang akan mengaktifkan motor putar arah.

Group: General Advanced

Name: **MOTORREVERSE**

Type:

- Group
- Digital
- Analog
- String

I/O Device: MASTER ST0

I/O Address: m007

Des:

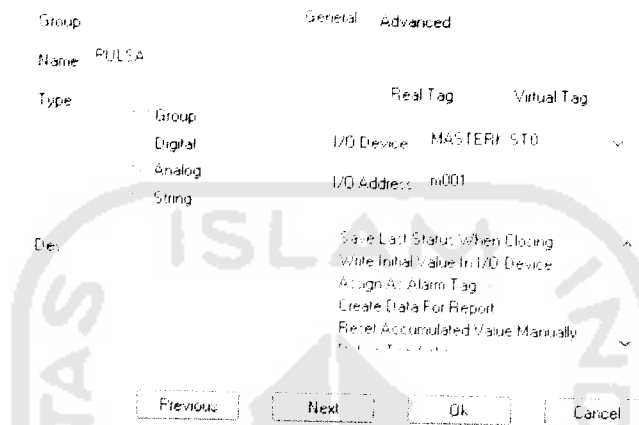
- Save Last Status When Closing
- Write Initial Value In I/O Device
- Reverse Value Read From I/O Device
- Assign As Alarm Tag
- Create Data For Report

Buttons: Previous, Next, Ok, Cancel

Gambar 3.12 Konfigurasi Pada Database MOTORREVERSE

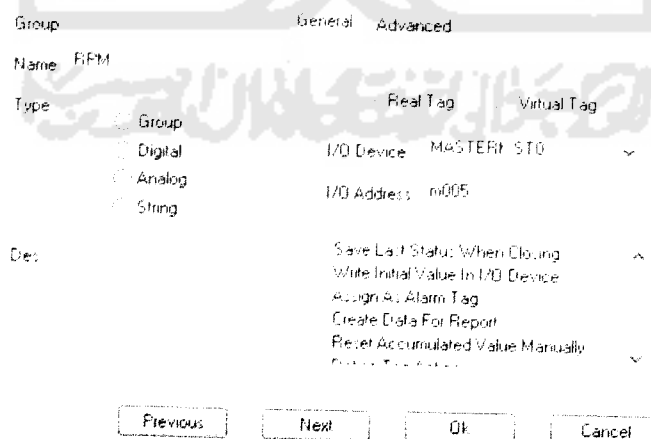
Database ketiga yang dibuat yaitu dengan nama PULSA. Pada database ini bertipe analog yang berarti aktif dengan inputan data. Pada database ini akan

menstransfer data pada alamat M001 yang merupakan alamat inputan periode pada instruksi PWM.



Gambar 3.13 Konfigurasi pada Database PULSA

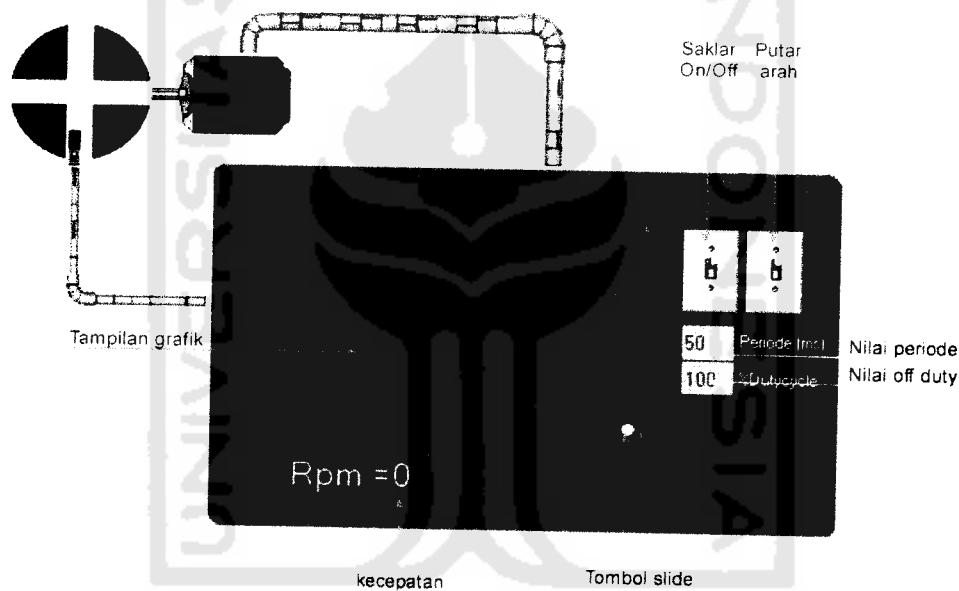
Database yang keempat yaitu dengan nama RPM. Pada database ini bertipe analog dan memanggil alamat memori M005. Alamat ini digunakan PLC sebagai alamat hasil perhitungan kecepatan dari instruksi *High Speed Counter* yang diperoleh dari pembacaan pulsa yang dihasilkan oleh *optocoupler*.



Gambar 3.14 Konfigurasi pada Database RPM

Pada database yang kelima yaitu dengan nama PERSEN. Pada database ini bertipe analog dan mentransfer data pada alamat memori M003 yang merupakan inputan dari % off duty cycle pada instruksi PWM.

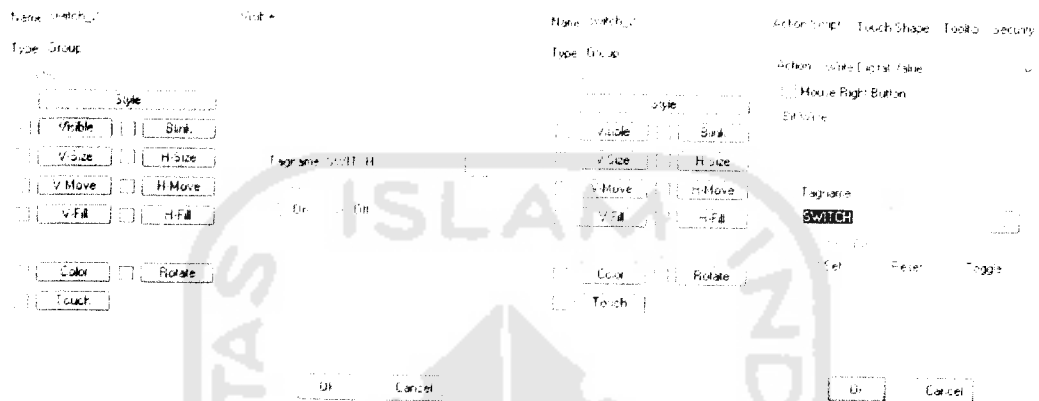
Setelah membuat database langkah selanjutnya adalah membuat tampilan simulasi seperti gambar berikut



Gambar 3.15 Monitoring Simulasi Kecepatan Motor DC

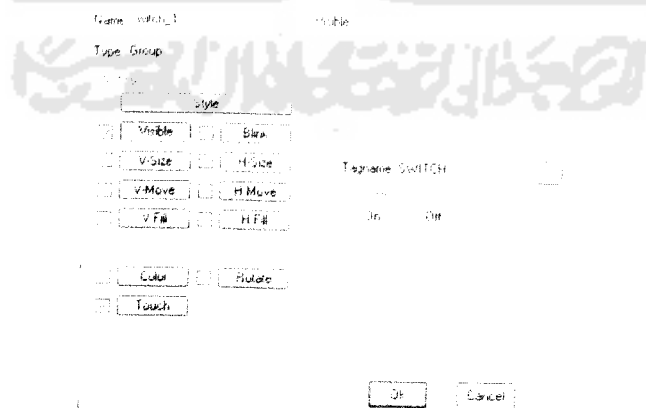
Pada gambar simulasi diatas terdiri dari beberapa bagian yaitu tombol saklar On/Off, tombol putar arah, tombol slide, tampilan input nilai periode, tampilan nilai Off duty, tampilan nilai kecepatan, dan tampilan grafik . Semua bagian tersebut memiliki konfigurasi yang berbeda.

Pada tombol saklar On/Off terdiri dari 2 tombol yaitu tombol On dan tombol Off yang memiliki konfigurasi sebagai berikut:



Gambar 3.16 Konfigurasi Pada Tombol Saklar Off

Konfigurasi tombol saklar Off diatas pada kolom tagname dipilih database SWITCH dengan kondisi Off. Untuk mengaktifkan saklar saat ditekan oleh mouse yaitu dengan memilih konfigurasi *Touch* dengan *action* yang dipilih adalah *Write Digital Value*.

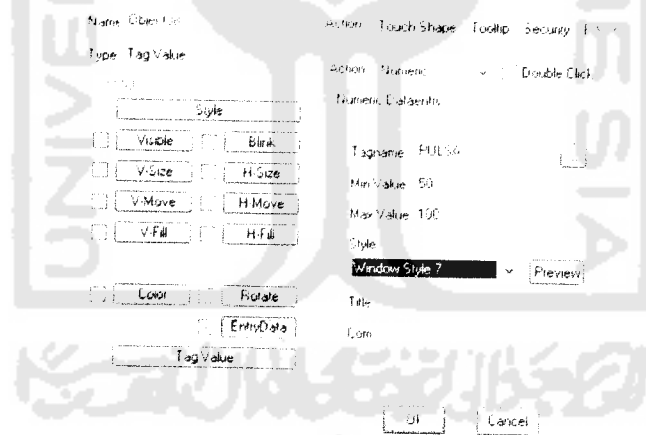


Gambar 3.17 Konfigurasi pada Tombol Saklar On

Konfigurasi tombol saklar On diatas untuk settingan *touch* dan tagname sama dengan saklar Off hanya kondisi yang dipilih adalah On. Tombol saklar On/Off ini berfungsi untuk mengaktifkan sistem secara real dengan memanggil database yang telah dibuat.

Pada tombol saklar putar arah konfigurasinya sama dengan saklar On/Off hanya database yang dipanggil adalah MOTORREVERSE yang akan mengaktifkan motor putar arah saat saklar itu ditekan.

Untuk membuat inputan pada nilai periode dilakukan dengan membuat dynamic tag pada tool draw dengan konfigurasi sebagai berikut :

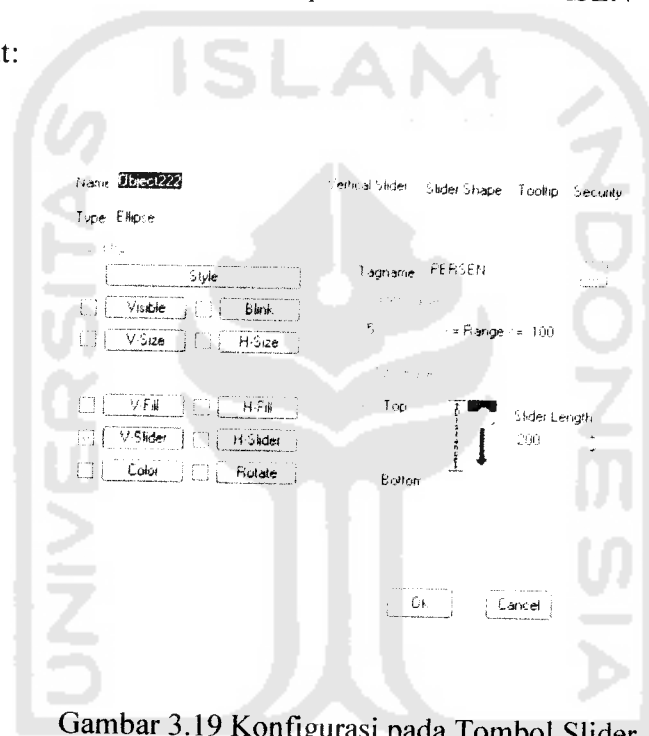


Gambar 3.18 Konfigurasi pada Inputan Nilai Periode

Konfigurasi diatas dipilih *EntryData* karena pada bagian ini berfungsi untuk memberikan dan menstransfer data pada database PULSA yang digunakan sebagai

nilai periode pada instruksi PWM. Nilai inputan yang digunakan untuk inputan data antara 0 – 100%.

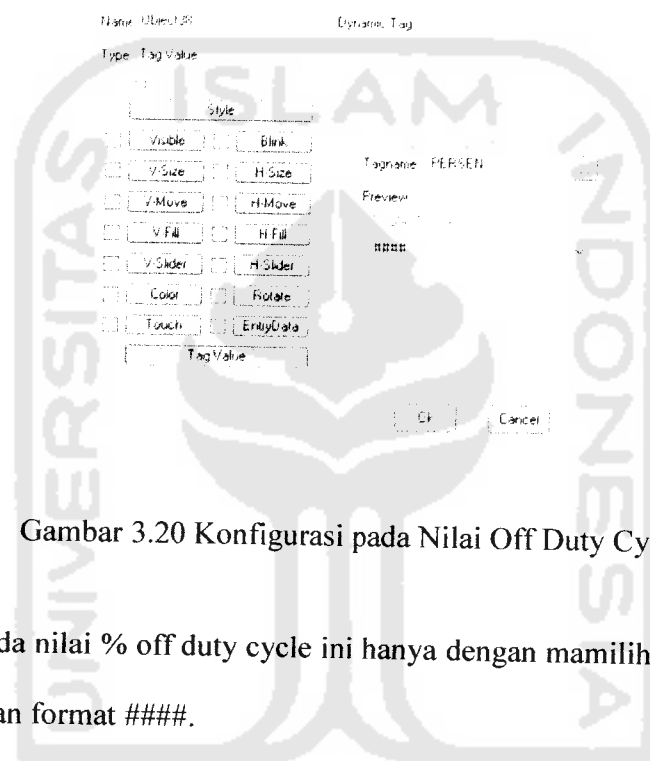
Tombol Slide berfungsi sebagai pengatur nilai inputan pada % *off duty* cycle. Nilai tersebut kemudian ditransfer pada database PERSEN dengan konfigurasi sebagai berikut:



Gambar 3.19 Konfigurasi pada Tombol Slider

Konfigurasi slider yang dipilih adalah V-Slider yang berarti akan bergeser secara vertical. Database yang digunakan yaitu PERSEN dengan range 5 – 100. Pada bagian *slider base* dipilih top agar pada saat bergeser ke bawah kecepatan motor semakin menurun. Untuk ukuran slide diisi sampai 200.

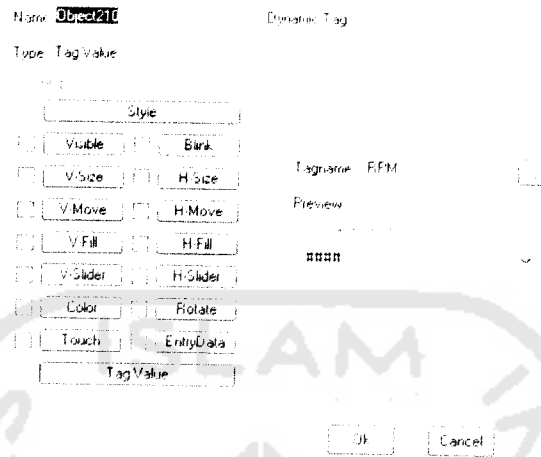
Untuk menampilkan data % *off duty cycle* dari tombol *slider* dilakukan dengan membuat *dynamic tag* seperti pada inputan periode dengan konfigurasi sebagai berikut:



Gambar 3.20 Konfigurasi pada Nilai Off Duty Cycle

Konfigurasi pada nilai % *off duty cycle* ini hanya dengan mamilih *Tagname* database PERSEN dengan format #####.

Untuk menampilkan nilai kecepatan dari hasil pembacaan dan perhitungan pulsa pada instruksi *High Speed Counter* dilakukan dengan membuat *dynamic tag*. Konfigurasi yang dipilih sama dengan nilai *off duty cycle* hanya saja *Tagname* yang dipilih adalah database RPM.



Gambar 3.21 Konfigurasi pada Nilai Kecepatan

Dari hasil perhitungan kecepatan yang dihasilkan dapat ditampilkan dengan grafik menggunakan tool *trend* dengan konfigurasi sebagai berikut :



Gambar 3.22 Konfigurasi pada Tampilan Grafik

Pada konfigurasi diatas *Trend Type* yang dipilih adalah YT yang menunjukkan grafik hubungan antara y axis terhadap waktu. Y axis disini dapat dinyatakan dengan nilai kecepatan. Untuk memasukan nilai kecepatan dapat dipilih *Tagname* pada database RPM. Nilai kecepatan dapat berubah seiring dengan perubahan waktu dalam inputan data.

