

# Perancangan HMI (*Human Machine Interface*) Untuk Pengendalian Kecepatan Motor DC

Heri Haryanto<sup>1</sup>, Sarif Hidayat<sup>2</sup>

<sup>1,2</sup>Jurusan Elektro Fakultas Teknik Universitas Sultan Ageng Tirtayasa  
Jl. Jendral Sudirman KM. 03 Cilegon – Banten 42435  
email : elektrojos@yahoo.com

**Abstrak** — HMI (*Human Machine Interface*) adalah sistem yang menghubungkan antara manusia dan mesin. Sistem HMI berupa *Graphic user interface (GUI)* pada suatu tampilan layar komputer yang akan dihadapi oleh operator mesin atau pengguna yang akan memonitoring dan mengendalikan kecepatan motor DC dengan metode PID secara *real time*. Penelitian dilakukan dengan merancang sebuah sistem HMI menggunakan software Labview 2009. Hasil pengujian sistem monitoring pengendalian motor DC pada fungsi tombol knop dan tombol on/off yang divisualisasikan sudah berperan sesuai dengan fungsinya dengan waktu sampling 70 ms. Kendali PID yang sesuai dengan karakteristik sistem yang telah dibuat adalah  $K_p = 1$   $K_i = 4$   $K_d = 0.3$  dengan *time delay* 0,49 detik, *rise time* sebesar 1,99 detik, *settling time* sebesar 2,24 detik, dan *steady state error* sebesar 1,76% .

**Kata Kunci** : HMI, Motor DC, Kendali PID, Labview

**Abstract** — HMI (*Human Machine Interface*) is a system that links between humans and machines. HMI system in the form (*GUI*) on a computer display screen that will be faced by machine operators or users who will monitor and control the speed of a DC motor with PID methods in *real time*. Research carried out by designing a system using the software Labview HMI 2009. Research results obtained of DC motor control monitoring system on the function knob and buttons on / off button is a role that was visualized according to function with time sampling 70 ms. At the appropriate PID control system with the characteristics that have been made is  $K_p = 1$   $K_i = 4$   $K_d = 0.3$  with a time delay of 0.49 seconds, rise time of 1.99 seconds, settling time of 2.24 seconds, and the steady-state error for 1.76%.

**Keywords** : HMI, Motor DC, PID control, Labview

## I. PENDAHULUAN

Tujuan dari HMI (*Human Machine Interface*) adalah untuk meningkatkan interaksi antara mesin dengan operator melalui tampilan layar komputer sehingga memenuhi kebutuhan pengguna terhadap informasi sistem yang diberikan sehingga mempermudah pekerjaan fisik. HMI dapat berupa pengendalian dan visualisasi komputer yang bersifat *real time*. Suatu sistem bekerja dengan pemantauan langsung membutuhkan pekerja yang selalu siap memberikan informasi secara cepat, tepat dan handal tetapi hal ini tidak dapat dilakukan oleh manusia. Sesuai dengan tujuan dan tugas HMI maka, HMI dalam industri sangat penting peranannya dalam sistem *monitoring* dan kendali suatu sistem produksi sehingga dengan sistem ini dapat menghemat waktu dan tenaga kerja untuk pengamatan dan pengendalian setiap stasiun kerja produksi. Pada penelitian ini, dirancang sebuah sistem HMI (*Human Machine Interface*) dengan bantuan software Labview 2009 untuk kendali kecepatan motor DC secara *real time*. Sistem HMI yang dirancang berupa tampilan *Graphic User Interface (GUI)* pada suatu tampilan layar komputer yang akan dihadapi oleh operator mesin (pengguna) yang akan mengamati dan mengendalikan kecepatan motor DC. Kendali kecepatan motor DC menggunakan metode PID sehingga meningkatkan kestabilan kecepatan motor tersebut. Hasil respon kendali PID terhadap kecepatan motor DC ditampilkan dengan grafik.

Pada Penelitian ini bertujuan merancang HMI untuk memonitoring sistem pengendalian kecepatan putaran motor DC, sehingga dapat diketahui karakteristiknya.

Karena luasnya kajian penelitian, maka batasan masalah yang diberlakukan dalam penelitian ini adalah:

1. Desain HMI diimplementasikan dengan software Labview 2009.
2. Variabel yang dikendalikan adalah kecepatan.
4. *Plant dianggap dalam kondisi ideal* (tidak ada kerusakan) atau dalam kondisi normal.

## II STUDI PUSTAKA

### A. HMI (*Human Machine Interface*)

HMI (*Human Machine Interface*) adalah sistem yang menghubungkan antara manusia dan teknologi mesin. HMI dapat berupa pengendali dan visualisasi status baik dengan manual maupun melalui visualisasi komputer yang bersifat *real time*. Sistem HMI biasanya bekerja secara *online* dan *real time* dengan membaca data yang dikirimkan melalui I/O port yang digunakan oleh sistem *controller*-nya. Port yang biasanya digunakan untuk *controller* dan akan dibaca oleh HMI antara lain adalah *port com*, *port USB*, *port RS232* dan ada pula yang menggunakan *port serial*. Tugas dari HMI (*Human Machine Interface*) yaitu membuat visualisasi dari teknologi atau sistem secara nyata. Sehingga dengan desain HMI dapat disesuaikan sehingga memudahkan pekerjaan fisik. Tujuan dari HMI adalah untuk meningkatkan interaksi antara mesin dan operator melalui tampilan layar komputer dan

memenuhi kebutuhan pengguna terhadap informasi sistem. HMI dalam industri *manufacture* berupa suatu tampilan GUI (*Graphic User Interface*) pada suatu tampilan layar komputer yang akan dihadapi oleh operator mesin maupun pengguna yang membutuhkan data kerja mesin. HMI terdapat berbagai macam visualisasi untuk *Monitoring* dan data mesin yang terhubung secara *online* dan *real time*. HMI akan memberikan suatu gambaran kondisi mesin yang berupa peta mesin produksi dapat dilihat bagian mesin mana yang sedang bekerja. Pada HMI juga terdapat visualisasi pengendali mesin berupa tombol, *slider*, dan sebagainya yang dapat difungsikan untuk mengendalikan mesin sebagaimana mestinya. Selain itu dalam HMI juga ditampilkan *alarm* jika terjadi kondisi bahaya dalam sistem. Sebagai tambahan, HMI juga menampilkan data-data rangkuman kerja mesin termasuk secara grafik.

**B. Pengaturan Kecepatan Motor DC [9]**

Untuk mengatur kecepatan motor dipergunakan persamaan

$$n = \frac{V_t - I_a \cdot R_a}{k \cdot \phi} \quad (1)$$

Dimana :

$n$  = Kecepatan Motor

$V_t$  = Tegangan Terminal motor

$I_a$  = Arus Jangkar

$R_a$  = Tahanan Jangkar

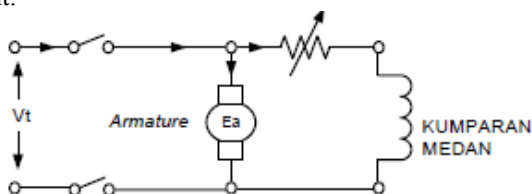
$k$  = Konstanta motor

$\phi$  = Fluk Medan

Berdasarkan persamaan (1), kecepatan motor dapat diatur dengan mengatur tegangan terminal, tahanan kumparan jangkar, tegangan nilai fluks.

Pengaturan tersebut dapat dilakukan sebagai berikut :

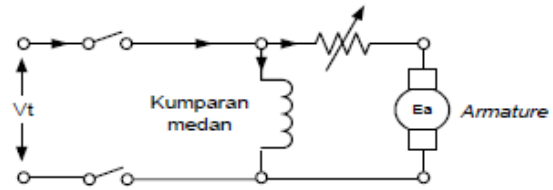
1. Pengaturan kecepatan dengan mengatur medan (*fluks*) Caranya dengan menyisipkan tahanan variabel yang dipasang secara seri terhadap kumparan medan (pada motor *shunt*), sehingga dapat diatur arus medan dan *fluks*-nya. Cara ini sangat sederhana, murah, dan rugi panas yang ditimbulkannya kecil. Besarannya *fluks* yang dapat dicapai oleh kumparan medan terbatas, sehingga kecepatan yang dapat diatur terbatas. Motor yang biasa diatur dengan cara ini adalah motor shunt atau motor kompon. Pada gambar 1 memperlihatkan pengaturan kecepatan motor dengan mengatur Medan shunt.



Gambar 1. Pengaturan Kecepatan dengan Mengatur Medan Shunt

2. Pengaturan kecepatan dengan mengatur tahanan jangkar Caranya dengan menyisipkan tahanan variabel secara seri terhadap tahanan jangkar, sehingga tahanan jangkar juga dapat diatur yang berarti kecepatan motor

juga dapat diatur. Cara ini jarang dipakai karena penambahan tahanan seri terhadap tahanan jangkar menimbulkan rugi panas yang besar, rangkaian pengaturannya diperlihatkan pada gambar 2 di bawah ini :



Gambar 2. Pengaturan Kecepatan dengan Mengatur Tahanan Jangkar 12

3. Pengaturan kecepatan dengan mengatur tegangan terminal Caranya dengan mengatur tegangan terminal motor sehingga kecepatan motor dapat diatur. Kecepatan motor akan bertambah jika menaikkan sumber tegangan dan kecepatan motor akan berkurang jika menurunkan sumber tegangan. Cara ini menghasilkan pengaturan kecepatan motor yang sangat halus dan banyak digunakan dalam kehidupan sehari-hari.

**C. Karakteristik Kendali PID [8]**

Kendali PID terdiri dari tiga jenis cara pengaturan yang saling dikombinasikan yaitu kendali proporsional, kendali integral dan kendali derivatif.

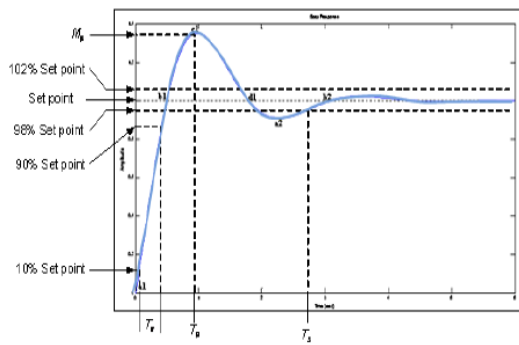
Masing-masing memiliki parameter tertentu yang harus diset untuk dapat bekerja dengan baik, yang disebut sebagai konstanta,  $K_p$  adalah konstanta proporsional,  $K_i$  menyatakan konstanta integral dan  $K_d$  menyatakan konstanta derivatif. Secara umum persamaan kendali PID adalah seperti pada persamaan 2 :

$$U(s) = K_p \cdot E(s) + \frac{K_i}{s} \cdot E(s) + K_d \cdot s \cdot E(s) \dots \quad (2)$$

Kendali proporsional akan memberikan efek mempercepat respon, tetapi tidak menghapus kesalahan keadaan tunak. Kendali integral akan memberikan efek menghapus kesalahan keadaan tunak, tetapi berakibat memburuknya respon transient. Kendali derivatif akan memberikan efek meningkatnya stabilitas sistem serta mengurangi *overshoot*. Efek dari setiap kendali ( $K_p$ ,  $K_i$ ,  $K_d$ ) dalam sistem *close loop* memiliki kelebihan dan kekurangan dapat diperlihatkan pada tabel 1

Tabel 1 Karakteristik Kendali PID

	Waktu Naik	Overshot	Waktu Penetapan	Kesetabilan
Pembesaran $K_p$	Berkurang	Bertambah	Sedikit bertambah	Menurun
Pembesaran $K_i$	Sedikit berkurang	Bertambah	Bertambah	Menurun
Pembesaran $K_d$	Sedikit berkurang	Berkurang	Berkurang	Meningkat



Gambar 3. Kurva Respon Proses

Parameter unjuk kerja pada gambar 3 seperti waktu naik, waktu puncak, persen lonjakan dan waktu penatap dapat dijelaskan sebagai berikut :

- Waktu naik ( $T_r$ ): waktu yang diperlukan respon untuk naik dari 10% – 90%
- Waktu puncak ( $T_p$ ): waktu yang diperlukan respon untuk mencapai lonjakan maksimum ( $M_p$ ).
- Persen lonjakan : lonjakan maksimum ( $M_p$ ) yang dialami oleh respon.
- Waktu penatap ( $T_s$ ): waktu yang diperlukan respon untuk mencapai dan menetap disekitar 95%-98% dari set point.

D. Konsep Dasar Labview

National Instruments LabVIEW adalah bahasa pemograman computer yang berbasis grafik. Pemograman Lab VIEW tidak menggunakan basis text seperti pada Visual Basic atau Delphi misalnya. Dengan memakai pemograman grafik ini maka membangun suatu aplikasi akuisisi data dan instrumentasi/control menjasi lebih mudah dan cepat. LabVIEW digunakan untuk membangun berbagai aplikasi akuisii data dan system instrumentasi dan control. LabVIEW dapat dihubungkan dengan hardware (perangkat keras) buatan National Instruments seperti data akuisisi, image akuisisi, motion control dan input/output untul aplikasi pengendalian industry dan lain lain. LabVIEW dapat digunakan untuk menghubungkan dengan aplikasi lain melalui ActiveX, Web, DLL dan Shared Library dan berbagai jenis protocol lainnya. LabVIEW dapat digunakan secara intensif untuk berbagai aplikasi industri seperti telekomunikasi, manufaktur, automotive, semikonduktor, biomedical,aerospace, electronics. Aplikasi pada bidang bidang ini mencakup semua tahap seperti research and development, engineering and validation, manufacturing, test dan service.

LabVIEW dengan mudah dapat dihubungkan dengan peralatan kontrol industri seperti Programmable Logic Control (PLC), dan Distributed Control System (DCS). []

Labview merupakan salah satu dari sekian banyak sarana pemograman komputer. Seperti halnya sarana pemograman lainnya Labview dikembangkan untuk perancangan dan rekayasa data sehingga dapat menampilkan dan memproses segala macam fungsi untuk melakukan manipulasi terhadap fungsi yang diinginkan. Labview memiliki 2 ruang kerja, pada

masing-masing ruang kerja memiliki antarmuka grafik tersendiri. Setiap simbol grafik ini mewakili banyak kata/perintah yang digunakan dalam bahasa teks, dengan demikian waktu yang diperlukan dalam perancangan program yang dilakukan oleh seorang pemrogram akan menjadi lebih efisien. Salah satu keunggulan dari Labview adalah aliran pemrograman yang dapat diamati proses kerjanya, sehingga jika terjadi kesalahan dalam pengolahan data dapat diketahui dengan mengamati proses tersebut.

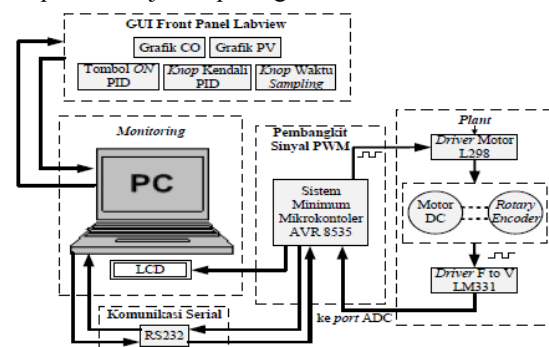
Program ini dirancang khusus untuk membuat gambaran/simulasi kerja suatu instrumen industri, komunikasi data, akuisisi data, sistem kendali, perancangan dan perhitungan matematika.

Labview menyediakan *tool* untuk mengolah objek dan melakukan konfigurasi terhadap nilai dan konstanta suatu objek serta digunakan untuk menghubungkan atau menyusun bahasa grafik yang digunakan. Pemrograman Labview telah dikelompokkan dalam masing-masing ruang. Pada *front panel* disediakan *control pallete* yang digunakan sebagai penampil data I/O. Pada bagian *block diagram* disediakan *function pallete* yang digunakan sebagai pengolah I/O data. *Icon* dan *connector panel* digunakan untuk mengidentifikasi VI (*Virtual Instrument*) sehingga bisa digunakan untuk VI yang lain.

BAB III METODOLOGI PENELITIAN

A. Sistem HMI

Sistem HMI yang akan dirancang dalam penelitian ini akan digunakan sebagai pengendali motor DC dengan metode PID dengan menggunakan software Labview. Sistem yang akan dibangun terdiri dari perangkat keras berupa plant motor DC dan interface berupa sistem mikrokontroler. Sedangkan untuk keperluan pengendalian dan monitoringnya dibangun perangkat lunak dengan menggunakan labview. Secara keseluruhan sistem yang akan dirancang pada penelitian ini seperti ditunjukan pada gambar 4



Gambar 4. Blok Diagram HMI Sistem Kendali Kecepatan Motor DC

Adapun penjelasan blok diagram sistem kendali kecepatan motor DC adalah sebagai berikut:

1. Sistem minimum mikrokontroler AVR ATmega8535 merupakan sebagai pembangkit sinyal PWM (*Pulse With Modulation*), menerima data dari

driver *F to V*, mengirim data ke LCD, mengirim data dan menerima data ke PC (*Personal Computer*) melalui RS232.

2. LCD (*Liquid Crystal Display*) digunakan untuk menampilkan data *setting point* dan keluaran kecepatan motor DC.

3. PC (*Personal Computer*) untuk menampilkan hasil keluaran *rotary encoder* dalam bentuk grafik dan pengendalian kecepatan motor DC melalui *setting point* pada *control panel* Labview .

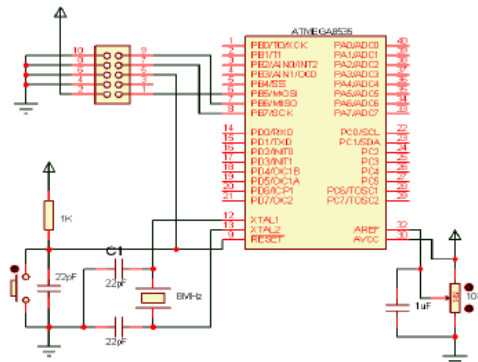
4. Catu daya (*power supply*) digunakan untuk memberikan catu daya bagi rangkaian sistem minimum mikrokontroler AVR ATmega8535, *driver* motor, *driver F to V* dan motor DC.

**B. Sistem Minimum Mikrokontroler AVR 8535**

Pada penelitian ini, menggunakan sistem minimum mikrokontroler AVR ATmega8535 sebagai pembangkit sinyal PWM (*Pulse Width Modulation*), yang mana sinyal PWM ini digunakan untuk pengendalian kecepatan pada motor DC.

Sistem minimum adalah beberapa rangkaian dasar yang digunakan untuk mengaktifkan sebuah mikrokontroler AVR ATmega8535. Mikrokontroler AVR ATmega8535 memiliki isolator *on-chip*, yang dapat digunakan sebagai sumber detak (*clock*) ke CPU. Sumber detak (*clock*) diperoleh dengan sebuah *crystal* 27 sebesar 8 MHz dan dua buah kapasitor sebesar 22pF sebagai penstabil yang dihubungkan dengan *pin* XTAL1 dan XTAL2 pada mikrokontroler AVR ATmega8535. Sedangkan rangkaian *reset* berfungsi untuk me-*reset* program pada mikrokontroler AVR ATmega8535. *Reset* tersebut diperoleh dengan prinsip menghubungkan *pin reset* mikrokontroler (*pin* 9) dengan logika 1 (*high*) atau 5V. *Pin reset* membutuhkan logika 1 (*high*) agar mikrokontroler AVR ATmega8535 melakukan proses *reset* (pengeksekusian ulang program dari awal program dengan alamat 0x000H). Proses akan berlangsung secara otomatis berdasarkan prinsip pengisian dan pengosongan muatan pada kapasitor. Selama pengisian muatan pada *pin reset* berada dalam kondisi *low*, setelah muatan penuh, kapasitor akan mulai melepaskan muatan dan akan menyalut *pin reset* sehingga terjadi proses *reset*. Dalam sistem minimum ini rangkaian *reset* di tambah dengan *pushbutton switch* agar pemakai dapat melakukan *reset* secara manual. Rangkaian sistem minimum mikrokontroler AVR ATmega8535 dapat ditunjukkan pada gambar 5.

Adapun penggunaan *port* pada sistem minimum mikrokontroler AVR ATmega8535 ditunjukkan pada tabel 2. Pada mikrokontroler AVR ATmega8535 memiliki 4 (empat) buah *port* terdiri dari *port* A digunakan pembacaan nilai ADC, *port* B digunakan untuk mengunduh program dan pembangkit sinyal PWM, *Port* C digunakan untuk LCD, *Port* D<sub>2</sub> - D<sub>7</sub> tidak dioperasikan.



Gambar 5. Sistem Minimum Mikrokontroler AVR ATmega8535

Tabel 2. *Port* Pada Sistem Minimum Mikrokontroler AVR ATmega8535

Nama <i>Pin</i>	Fungsi Lain	Jenis <i>Pin</i>	Penggunaan aplikasi
<b>PORT A</b>			
PA0	ADC0	I	<i>Pin</i> dari <i>driver F to V</i>
PA1	ADC1		Tidak dioperasikan
PA2	ADC2		Tidak dioperasikan
PA3	ADC3		Tidak dioperasikan
PA4	ADC4		Tidak dioperasikan
PA5	ADC5		Tidak dioperasikan
PA6	ADC6		Tidak dioperasikan
PA7	ADC7		Tidak dioperasikan
<b>PORT B</b>			
PB0	XCK/T0		Tidak dioperasikan
PB1	T1		Tidak dioperasikan
PB2	INT2/AEN0		Tidak dioperasikan
PB3	OC0/AEN1		<b>PWM (<i>Pulse With Modulation</i>)</b>
PB4	SS		Tidak dioperasikan
PB5	MOSI	I	Masukan <i>SPI Bus Master Output/Slave</i>
PB6	MISO	O	Keluaran <i>SPI Bus Master Input/Slave</i>
PB7	SCK	I/O	<i>SPI Bus Serial Clock</i>
<b>PORT C</b>			
PC0	SCL		Tidak dioperasikan
PC1	SDA		Tidak dioperasikan
PC2			<i>Pin</i> Db7 LCD
PC3			<i>Pin</i> Db6 LCD
PC4			<i>Pin</i> Db5 LCD
PC5			<i>Pin</i> Db4 LCD
PC6	TOSC1		<i>Pin</i> E LCD
PC7	TOSC2		<i>Pin</i> RS LCD
<b>PORT D</b>			
PD0	RXD	I	Masukan <i>US.ART</i>
PD1	TXD	O	Keluaran <i>US.ART</i>
PD2	INT0		Tidak dioperasikan
PD3	INT1		Tidak dioperasikan
PD4	OC1B		Tidak dioperasikan
PD5	OC1A		Tidak dioperasikan
PD6	ICP		Tidak dioperasikan
PD7	OC2		Tidak dioperasikan

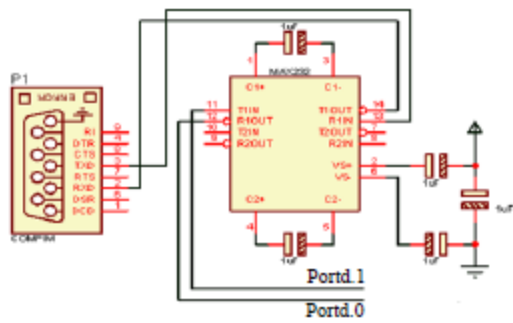
Keterangan :

1. Huruf yang di cetak tebal menandakan *port/pin* yang digunakan pada mikrokontroler AVR ATmega8535
2. I = *Input* (masukan)
3. O = *Output* (keluaran)

**C. Rangkaian Antarmuka Dengan PC**

Pada komputer terdapat *port* serial yang digunakan untuk komunikasi pada perangkat luar. Penggunaan *port* serial pada komputer dimaksudkan untuk dapat berkomunikasi dengan mikrokontroler AVR ATmega8535. Dalam hal ini, perlu adanya komponen tambahan berupa IC (*Integrated Circuit*) MAX232, yang merupakan komponen pengubah standar RS-232 ke standar TTL atau sebaliknya.

Adapun gambar 6 merupakan rangkaian antar muka dengan PC.



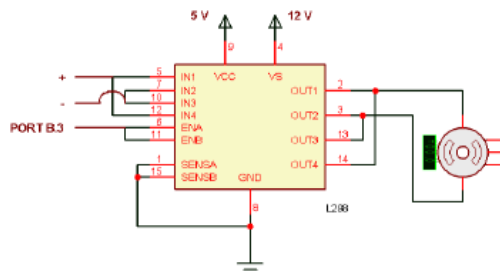
Gambar 6. Rangkaian Antarmuka Dengan PC

Pada gambar 6. rangkaian antarmuka dengan PC dihubungkan ke port D<sub>0</sub> dan D<sub>1</sub> Mikrokontroler AVR ATmega8535. Pada pin 11 IC MAX232 dihubungkan ke pin TDX (pin Portd.1) pada mikrokontroler, kemudian pin 12 IC MAX232 dihubungkan ke pin RXD (pin Portd.0) pada mikrokontroler. Sedangkan pin 13 IC MAX232 dihubungkan ke pin 3 (RX) dari port serial komputer dan pin 14 IC MAX232 dihubungkan ke pin 2 dari port serial komputer.

D. Rangkaian Driver Motor L298

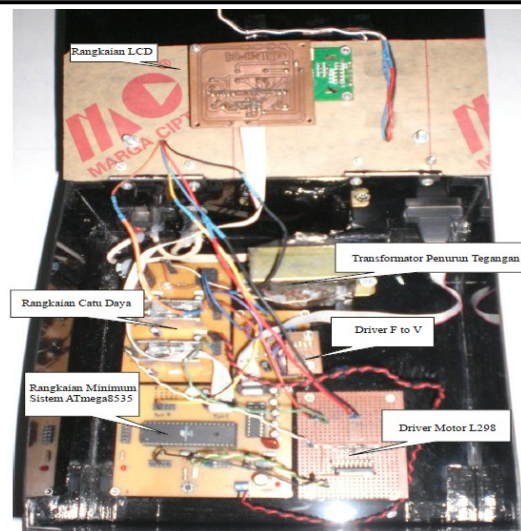
Sesuai dengan namanya, motor dc menggunakan tegangan DC (*Direct Current*). Putaran motor DC akan berbalik arah jika polaritas tegangan yang diberikan juga dirubah. Motor DC terdapat tegangan kerja yang bervariasi, ada yang memiliki tegangan 3V, 6V, 12V dan 24 V.

Pada perancangan driver motor digunakan IC L298 memiliki kemampuan menggerakkan motor DC sampai arus 2A namun dalam penggunaannya L298 dapat digunakan secara *parallel* sehingga kemampuannya menghantarkan arus menjadi 4 A. Rangkaian driver motor L298 ditunjukkan pada gambar 7.



Gambar 7. Rangkaian Driver Motor L298

Pada pin (*Enable*) dihubungkan ke port B.3 pada mikrokontroler AVR ATmega8535 dengan maksud driver motor L298 menerima sinyal PWM dari mikrokontroler untuk mengendalikan kecepatan motor DC tersebut. Gambar 8 dan 9 memperlihatkan hasil rancangan perangkat keras dari peralatan modul HMI



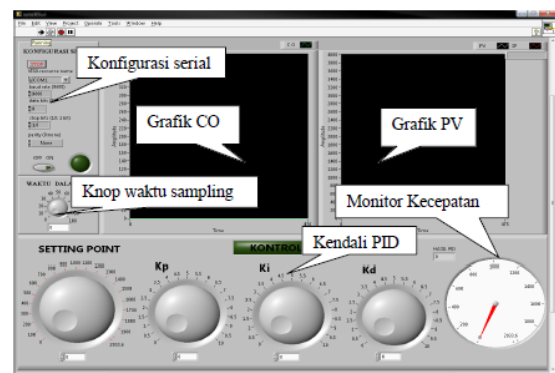
Gambar 8. Modul Rangkaian Pada Perancangan Alat



Gambar 9. Modul Motor yang akan dikendalikan

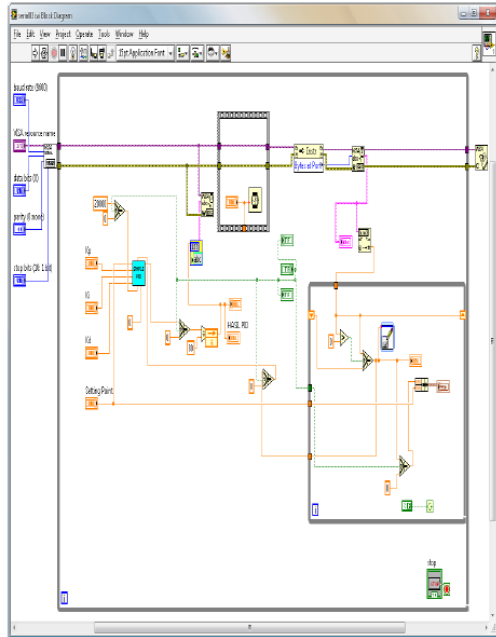
E. Perancangan HMI Sistem Monitoring Pengendalian Motor DC

Labview merupakan bahasa pemrograman dalam bentuk grafis yang banyak diadopsi dan digunakan untuk industri, akademis dan lab-lab pemerintah sebagai standar untuk program akuisisi data dan instrumentasi. Perangkat lunak ini menawarkan dalam penggunaan dan lebih atraktif serta memiliki fitur-fitur yang mudah di pahami. Pada perancangan ini, tampilan depan di desain pada *front panel* labview yang dapat ditunjukkan pada gambar 10.



Gambar 10. Tampilan Interface Pada Front Panel Labview

Pemograman grafis pada Labview ditampilkan digram blok yang ditunjukkan pada gambar 11. Terdapat sub program PID, program untuk menampilkan grafik, program untuk *setting* komunikasi serial dan program untuk pewaktuuan sinyal.

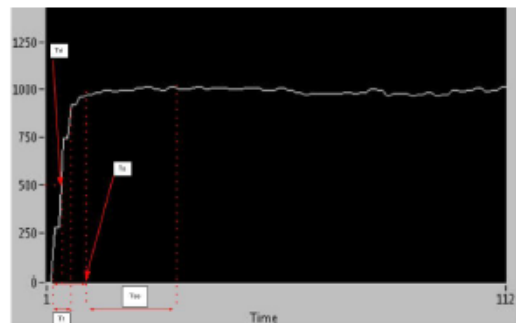


Gambar 11. Diagram Blok Pemograman PID Pada Labview

ada pada lampiran. Maka pada periode *input* 70 ms merupakan waktu sampling yang terbaik

B. Pengujian Sistem Kendali *Open Loop*

Pengujian sistem kendali *open loop ini* bertujuan mendapatkan karakteristik dari sistem yang telah dirancang sedemikian rupa. Langkah awal yang dilakukan adalah menghubungkan kabel serial dari komputer ke *plant*. Kemudian *plant* diberikan catu daya sehingga *plant* sudah siap untuk digunakan. Pada program Labview yang telah dirancang diberikan sebuah *control output* ke sistem (*plant*) sebesar 50 dengan pemberian waktu *sampling* 70 ms. Hasil respon sistem kendali *open loop* berupa grafik dapat ditunjukkan pada gambar 11.



Gambar 11. Respon Sistem Kendali *Open Loop*

BAB IV. HASIL DAN PENGUJIAN

A. Pengujian waktu *sampling*

Pengujian waktu *sampling* dilakukan dengan tujuan untuk mengetahui berapa besar pengaruh waktu *sampling* terhadap sistem kendali PID. Pengujian ini dilakukan dengan cara memutar tombol *knop* waktu searah jarum jam dengan nilai 2000 ms, 1000 ms, 500 ms, 250 ms, 100ms, 70 ms, 60 ms dengan nilai parameter  $K_p=1$   $K_i=4$   $K_d=0.3$ . Adapun tombol *knop* waktu sebelum di putar searah jarum jam. Hasil pengujian ini terdapat pada tabel 3.

Tabel 3. Hasil Pengujian waktu *sampling*

Periode <i>Input</i> (ms)	Frekuensi <i>Input</i> (Hz)	Keterangan
2000	0,5	Berosilasi
1000	1	Overshoot
500	2	Sistem lambat
250	4	Sistem lambat
100	10	Sistem bekerja
70	14,285	Sistem bekerja dengan baik
60	16,667	Berosilasi

Hasil pengamatan tabel 3 dengan melakukan pendekatan kriteria *Nyquist* yaitu frekuensi *sampling* minimum adalah dua kali sinyal analog yang dikonversi. dengan nilai frekuensi *sampling* minimum yang dihasilkan 30,52 Hz. Perhitungan frekuensi *sampling* minimum dan hasil gambar pengujian waktu *sampling*

Waktu respon pada gambar 11 adalah siklus pada grafik dikalikan dengan waktu *sampling*. Dari respon yang terlihat pada gambar 11 diperoleh karakteristik dan spesifikasi waktu tunda (*delay time*) sebesar 315 ms, waktu naik (*rise time*) sebesar 385 ms, waktu penetapan  $\pm 5\%$  (*settling time*), kesalahan keadaan tunak (*steady state error*) sebesar 5%.

Hasil respon sistem kendali *open loop* pada gambar 11, *control output* tidak dipengaruhi oleh sistem (umpan balik *plant*), sinyal keluaran (respon) yang dihasilkan tidak mempengaruhi sinyal masukan pada sistem sehingga nilai yang telah ditetapkan belum dapat dijadikan sebagai acuan seberapa optimalnya system yang telah dirancang.

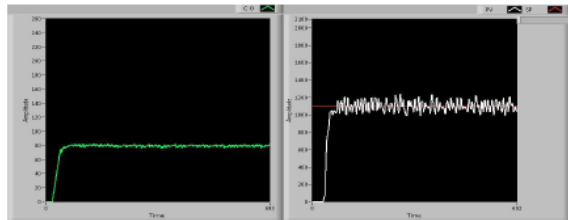
C. Pengujian Sistem Kendali PID

Pengujian sistem kendali PID dilakukan dengan tujuan mendapatkan nilai  $K_p$ ,  $K_i$ ,  $K_d$  yang sesuai dengan sistem. Langkah awal yang dilakukan adalah menghubungkan kabel serial dari komputer ke *plant*, kemudian *plant* diberikan catu daya sehingga *plant* sudah siap untuk digunakan.

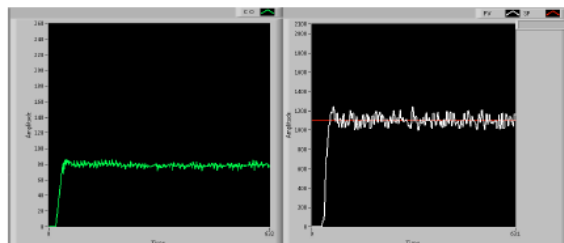
Proses selanjutnya memberikan waktu *sampling* sebesar 70 ms. Langkah berikutnya, pemberian nilai  $K_p = 1$ ,  $K_i = 3$   $K_d$  sebesar 0,1 sampai dengan 0,4. Nilai  $K_p$ ,  $K_i$ , dan  $K_d$  dikombinasikan dengan tujuan mendapatkan parameter yang sesuai dengan sistem. Setelah pemberian nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang dikombinasikan selanjutnya, menekan tombol *ON* untuk mengaktifkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang diberikan ke sistem. Setelah diperoleh hasil maka, tombol *OFF* harus ditekan dengan

tujuan menonaktifkan nilai  $K_p$  dan  $K_i$  yang diberikan ke sistem.

Gambar 12 merupakan grafik respon kendali PID dengan parameter  $K_p = 1$   $K_i = 3$   $K_d = 0,3$  diperoleh karakteristik dan spesifikasi waktu tunda (*delay time*) sebesar 0,98 detik, waktu naik (*rise time*) sebesar 2,17 detik, waktu penetapan (*settling time*) sebesar 4,41 detik, kesalahan keadaan tunak (*steady state error*) sebesar 2,54%.



Gambar 12. Respon Kendali PID dengan Nilai  $K_p = 1$   $K_i = 3$   $K_d = 0,3$ .



Gambar 13. Respon Kendali PID dengan Nilai  $K_p = 1$   $K_i = 4$   $K_d = 0,3$ .

Pada gambar 13 merupakan grafik respon kendali PID dengan parameter  $K_p = 1$   $K_i = 4$   $K_d = 0,3$  diperoleh karakteristik dan spesifikasi waktu tunda (*delay time*) sebesar 0,77 detik, waktu naik (*rise time*) sebesar 0,95 detik, waktu penetapan (*settling time*) sebesar 1,99 detik, kesalahan keadaan tunak (*steady state error*) sebesar 2,14%.

Hasil pengamatan respon kendali PID pada dengan konstanta  $K_p = 1$   $K_i = 3$  dengan nilai konstanta derivatif sebesar 0,1 sampai dengan 0,4 didapatkan karakteristik PID yang sesuai adalah pada konstanta  $K_p = 1$   $K_i = 3$   $K_d = 0,3$ .

Sedangkan pada pengamatan respon kendali PID dengan nilai konstanta  $K_p = 1$   $K_i = 4$  dan  $K_d$  sebesar 0.1 sampai dengan 0.4 didapatkan hasil yang sesuai adalah pada  $K_p = 1$   $K_i = 4$  dan  $K_d = 0,4$ , dengan membandingkan hasil respon terbaik maka sistem yang sesuai adalah dengan  $K_p = 1$   $K_i = 4$  dan  $K_d = 0,3$ .

#### D. Pengujian Kecepatan Motor DC Pada HMI dan Plant

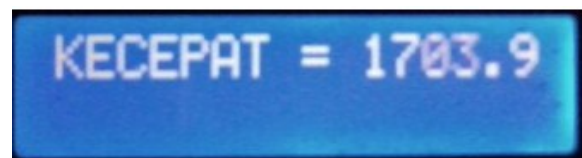
Pengujian kecepatan motor DC pada HMI dan *plant* bertujuan untuk mengetahui kondisi pemantauan pada layar komputer sama dengan kejadian pada *plant* sebenarnya. Langkah awal pengujian dilakukan dengan memberikan kecepatan secara bertahap kemudian diamati keadaan di *plant* dan layar komputer. Sebagai contoh dilakukan pengujian kecepatan motor 1700 rpm dengan cara memberikan sebuah nilai 1700 rpm pada

*setting point*. Pengamatan dapat dilakukan dengan melihat *speedometer* yang ditampilkan pada layar komputer, kemudian dilakukan pengamatan *plant* dengan melihat pada layar LDC. Gambar 14 menunjukkan hasil pengujian kecepatan motor DC pada HMI sebesar 1700 rpm.



Gambar 14. Kecepatan Motor 1700 rpm Pada HMI

Sedangkan gambar 15 menunjukkan hasil pengujian kecepatan motor DC pada *plant* sebesar 1700 rpm yang ditampilkan oleh LCD.



Gambar 15. Kecepatan Motor 1700 rpm Pada LCD

## BAB V. KESIMPULAN

Hasil perencanaan dan pembuatan sistem kemudian dilakukan pengujian, pengukuran dan analisa sistem yang telah dibuat didapat kesimpulan sebagai berikut :

1. Pada uji waktu sampling periode *input* 70 ms merupakan waktu *sampling* yang sesuai dengan *plant*.
2. Karakteristik kendali kecepatan motor DC pada HMI, Kendali PID pada parameter  $K_p = 1$   $K_i = 4$   $K_d = 0,3$  mampu untuk mengendalikan kecepatan motor dengan *time delay* sebesar 0,77 detik *rise time* sebesar 0,95 detik *settling time* sebesar 1,99 detik.
3. Pada pengujian beban bahwa semakin besar beban yang diberikan ke system maka kecepatan motor pun semakin berkurang, maka kontrol PID pun semakin lama untuk mencapai kesetabilan yang diinginkan.
4. Hasil pengujian kecepatan motor DC dengan HMI menunjukkan bahwa kecepatan motor yang ditampilkan pada HMI sudah sesuai dengan kondisi kecepatan pada *plant*.

### DAFTAR PUSTAKA

- [1] Darma Adiantoro, Perancangan Sistem Monitoring Chiller berbasis PC, Hasil hasil penelitian EBN tahun 2008.
- [2] Eka Budiono, “Programmable Automation Controller (PAC) dengan LabVIEW 7.1 terkoneksi Mikrokontroler dan PLC” , Penerbit Gava Media, Yogyakarta, 2009.

- 
- [3] Ibrahim, dogan. “*Microcontroller Based Applied Digital Control*” ,Department of Computer Engineering Near East University, Cyprus, 2006.
  - [4] Ogata, Katsuhiko., “*Modern Control Engineering*” Penerbit : Prentice – Hall, 1985.
  - [5] Prof. Jogiyanto. “*Pengenalan Komputer*”., Penerbit Andi , Yogyakarta , 2005.
  - [6] Retna Prasetya, Catur Edi Wibowo, “*Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0.* “ ,Penerbit Andi, Jogjakarta, 2004.
  - [7] Setiawan, Iwan “*Kontrol PID Untuk Proses Industri*” , PT Elex Media Komputindo 2008.
  - [8] Sugiarto, Indar , Desain dan Implementasi Modul DAQ Labview, Jurusan Teknik Elektro, 2008 Universitas Kristen Petra Surabaya, Jurnal
  - [9] Zuhail. “*Dasar Tenaga Listrik*”, Penerbit ITB Bandung, 1980.
  - [10] ST Microelectronic. “Dual Full Bridge Driver L298”. Datasheet Archive. Italy, 2008.