

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pemrograman *Software*

Proses yang dilakukan dalam perancangan *software* adalah melakukan pemrograman. Beberapa komponen yang perlu dilakukan pemrograman diantaranya adalah PLC dan HMI. Pemrograman PLC menggunakan *software CX Programmer* dan pemrograman HMI dilakukan dengan *software HMI Haiwel SCADA Cloud*.

4.1.1 Pemrograman CX - *Programmer*

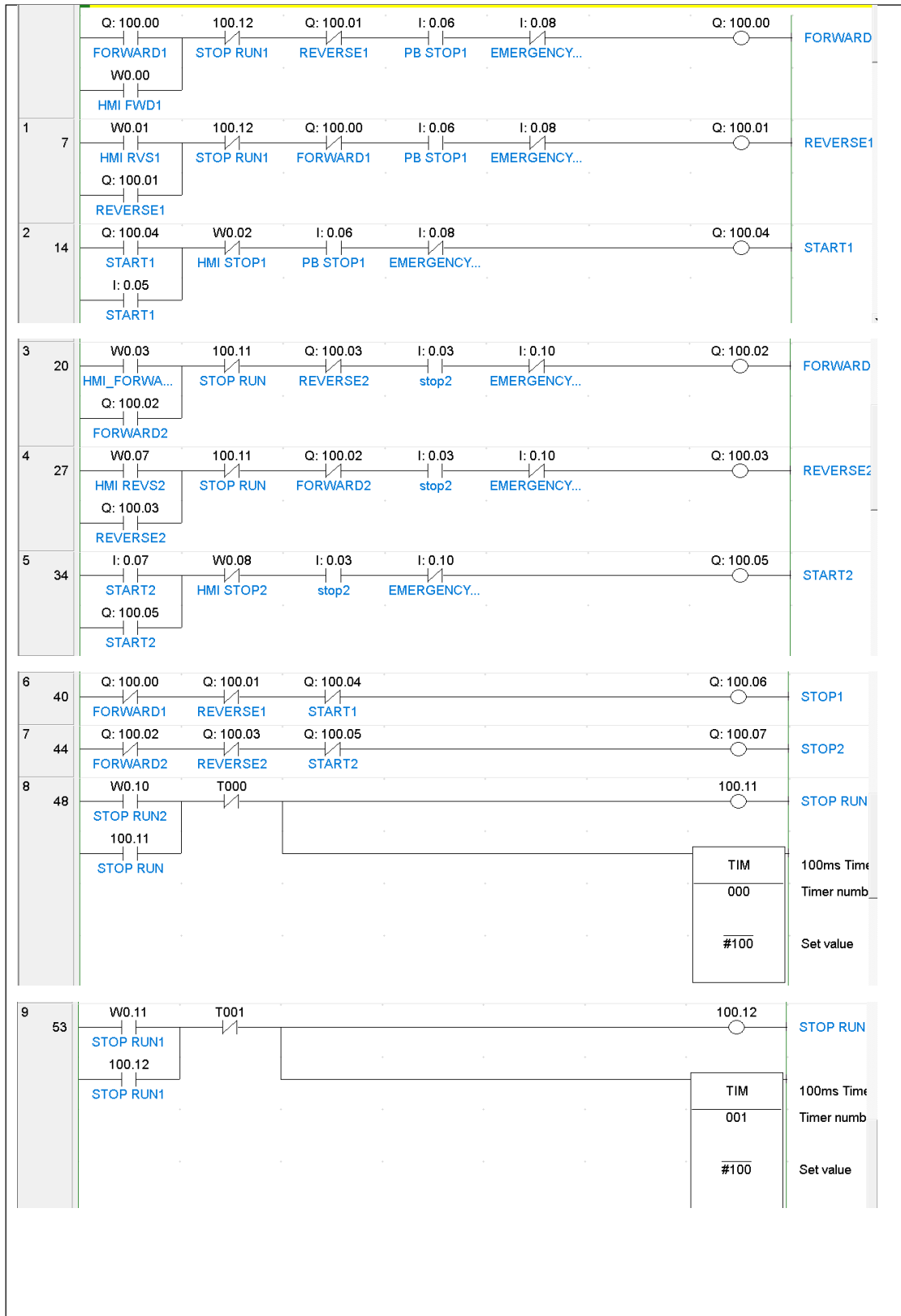
Untuk memberikan instruksi sistem pengoperasian PLC perlu melakukan pemrograman yang disesuaikan dengan sistem kerja alat, pemrograman tersebut dapat dilakukan dengan menggunakan perangkat lunak CX – *Programmer* yang sudah terinstal pada laptop atau PC.

Fungsi dari PLC pada alat ini adalah sebagai pengendali arah putaran motor listrik (*forward/reverse*). Selain itu juga PLC berfungsi untuk mengontrol *indikator lamp*, kontaktor dan *thermal overload relay* yang menjadi perangkat aktuator keluaran dari PLC.

Sinyal masukan dari PLC menggunakan *push button* yang berfungsi sebagai pemberi perintah kerja untuk mengoperasikan kontaktor. Pengalamatan dari *push button* disinkronkan agar dapat berfungsi sama dengan *push button* yang terdapat pada layar HMI, sehingga untuk mengaktifkan kontaktor dapat melalui *push button* dan juga layar HMI.

Keluaran PLC yang terhubung dengan kontaktor, TOR dan lampu indikator juga diberikan alamat yang disinkronkan dengan pengalamatan pada tampilan HMI agar dapat memberikan indikasi dari kondisi perangkat yang terhubung (kontaktor, TOR dan lampu indikator).

Ladder diagram dari pemrograman PLC yang telah dibuat menggunakan perangkat lunak CX – *Programmer* secara rinci dapat dilihat pada Gambar 4.1 di halaman selanjutnya.



Gambar 4. 1 Ladder Diagram Keseluruhan

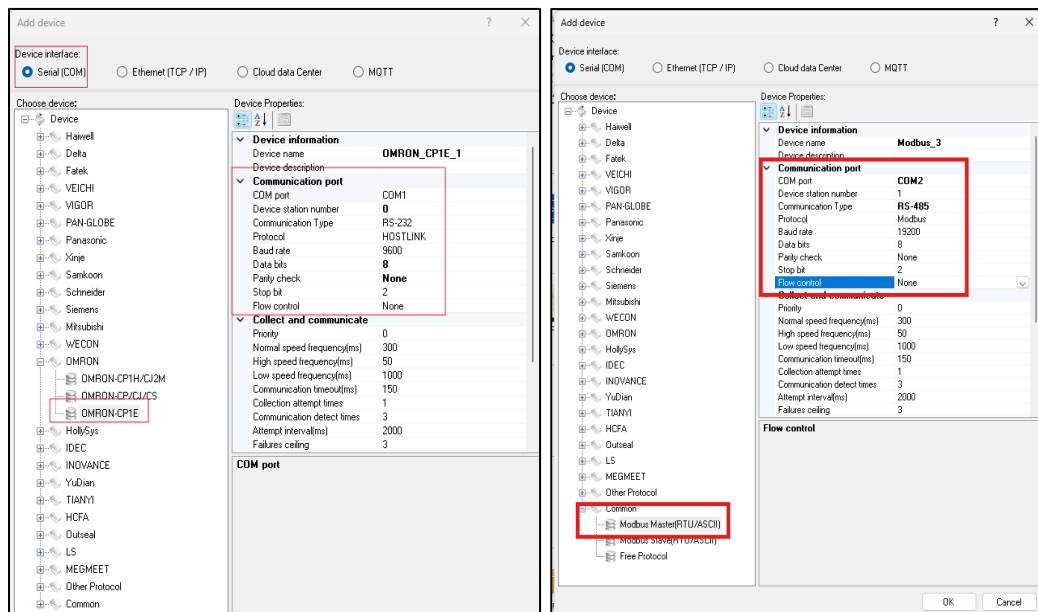
Name	Data Type	Address / Value	Rack Locati...	Usage	Comment
*	BOOL	0.00		In	PB FORWARD1
*	BOOL	0.02		In	stop1
*	BOOL	0.03		In	stop2
*	BOOL	0.05		In	START1
*	BOOL	0.06		In	PB STOP1
*	BOOL	0.07		In	START2
*	BOOL	0.08		In	EMERGENCY STOP1
*	BOOL	0.09		In	EMERGENCY STOP
*	BOOL	0.10		In	EMERGENCY STOP 2
*	BOOL	100.00		Out	FORWARD1
*	BOOL	100.01		Out	REVERSE1
*	BOOL	100.02		Out	FORWARD2
*	BOOL	100.03		Out	REVERSE2
*	BOOL	100.04		Out	START1
*	BOOL	100.05		Out	START2
*	BOOL	100.06		Out	STOP1
*	BOOL	100.07		Out	STOP2
*	BOOL	100.11		Work	STOP RUN
*	BOOL	100.12		Work	STOP RUN1
*	BOOL	W0.00		Work	HMI FWD1
*	BOOL	W0.01		Work	HMI RV1
*	BOOL	W0.02		Work	HMI STOP1
*	BOOL	W0.03		Work	HMI_FORWARD2
*	BOOL	W0.07		Work	HMI REVS2
*	BOOL	W0.08		Work	HMI STOP2
*	BOOL	W0.10		Work	STOP RUN2
*	BOOL	W0.11		Work	STOP RUN1

Tabel 4. 1 Address Function PLC

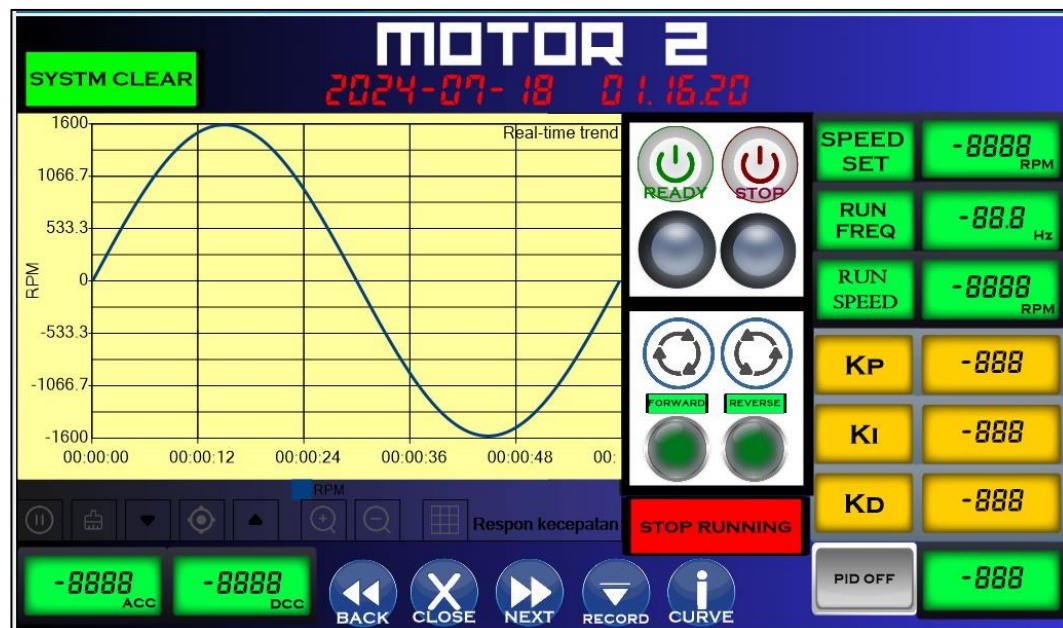
Selain *ladder diagram* juga dapat dilihat alamat setiap fungsi dari *CX Programmer* yang dapat dilihat pada tabel 4.1 di atas. Alamat ini nantinya disesuaikan dengan pemasangan *Wiring PLC* dan pemrograman *Haiwell SCADA Cloud*.

4.1.2 Pemrograman Haiwell Cloud SCADA

Pemrograman HMI dilakuakn terlebih dahulu dengan menentukan protokol komunikasi yang digunakan antara *device* dengan HMI. Pada Gambar 4.2 dapat dilihat protokol komunikasi HMI dengan PLC menggunakan RS 232 dan komunikasi HMI dengan *inverter* menggunakan RS 485.



Gambar 4. 2 Penentuan Protokol Komunikasi

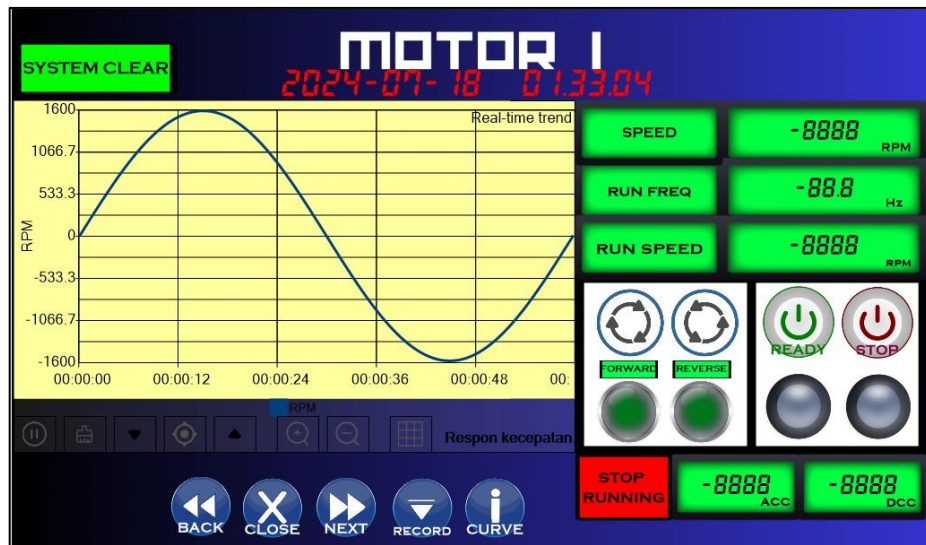


Gambar 4. 3 Hasil *Design* Tampilan Motor 1

Gambar 4.3 di atas merupakan hasil dari proses desain tampilan untuk sistem kontrol dan *monitoring* motor 2. Panel berwarna hijau merupakan panel yang berfungsi untuk mengatur data perintah dan memberikan data *monitoring* dari kecepatan, akselerasi, dekselerasi, frekuensi dan kecepatan yang sedang berjalan pada motor. Kemudian pada panel berwarna kuning berfungsi sebagai *setting* parameter PID. Pada panel yang berlatar belakang merupakan panel yang memberikan perintah kerja ke PLC yang nantinya pada PLC akan diteruskan ke *inverter*. *Setting* data masukan tersebut diantaranya perintah "ready" untuk mengaktifkan kontaktor, "stop" menonaktifkan kontaktor dan juga terdapat tombol untuk memberikan perintah *forward/reverse* motor.

Grafik pada tampilan ini menunjukkan nilai kecepatan motor Listrik secara *real time*. Kemudian panel berwarna biru merupakan tombol untuk memindahkan halman tampilan dari HMI.

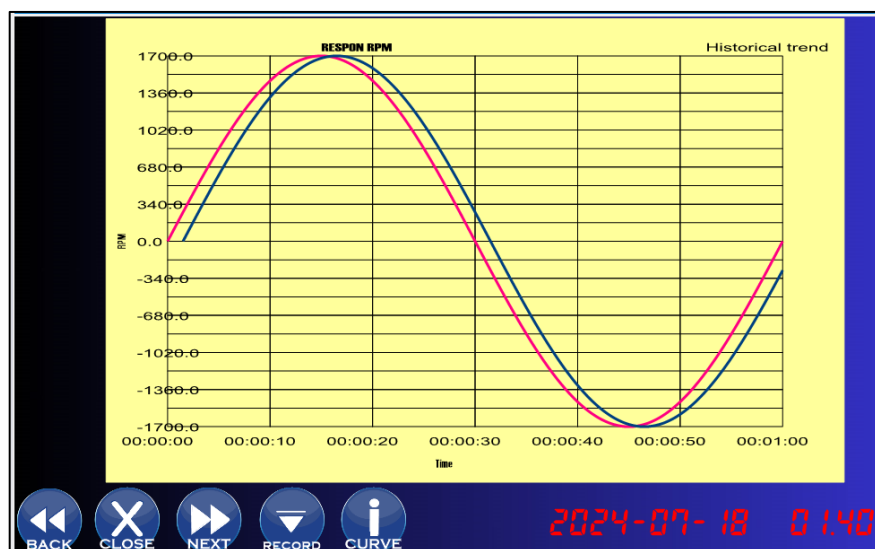
Panel bertuliskan "system clear" menunjukkan status *Thermal Overload Relay* dalam keadaan normal (tidak terjadi *overload*). Jika dalam keadaan sebaliknya maka status berubah menjadi "overload" dan latar belakangnya menjadi berwarna merah.



Gambar 4. 4 Tampilan *Monitor Motor 1*

Panel – panel pada *display* motor 1 juga memiliki fungsi yang sama dengan motor 2, hanya saja motor 1 tidak dilengkapi dengan *PID controller*. Tampilan *monitor* motor 1 dapat dilihat pada gambar 4.4 di atas.

Monitor Historical Trend dapat memberikan data berupa grafik kurva yang dapat ditentukan waktu memulai dan waktu berakhirnya proses perekaman data. Pada penelitian kali ini data yang diambil adalah data *respon* kecepatan motor terhadap perubahan parameter *PID*. Tampilan dari *monitor Historical Trend* dapat dilihat pada Gambar 4.5.



Gambar 4. 5 *Monitor Historical Trend*

DATA RECORD RESPON KECEPATAN DAN FREKUENSI				
Time	Variable1	Variable2	Variable3	Variable4
Content0	Content1	Content2	Content3	Content4

2024-07-18 02:00:12

Gambar 4. 6 Tabel Data Record

Gambar 4. 6 di atas merupakan tampilan dari *historical data* berupa tabel. Fungsinya sama dengan *historical Trend*, namun data yang ditampilkan berbentuk angka dalam tabel.

Gambar 4. 7 di bawah merupakan device PLC dengan *Variable name*, *Register type*, *Address format*, *Register Address* dan *Bit Address*. *Address Register* pada PLC device harus sesuai dengan *ladder diagram* PLC yang sudah dibuat sebelumnya.

	Variable name	Register type	Address format	Register address	Bit address	Address length	Data type	The mode of reading and writing	Collect frequency
1	HMI_FWD1	W(W)	Decimal		U	U	1 Bool	Read and write	Normal
2	HMI_RVS1	W(W)	Decimal		0	1	1 Bool	Read and write	Normal
3	HMI_STOP1	W(W)	Decimal		0	2	1 Bool	Read and write	Normal
4	HMI_FWD2	W(W)	Decimal		0	3	1 Bool	Read and write	Normal
5	HMI_RVS2	W(W)	Decimal		0	7	1 Bool	Read and write	Normal
6	HMI_STOP2	W(W)	Decimal		0	8	1 Bool	Read and write	Normal
7	START1	CIQ(CIO)	Decimal		0	5	1 Bool	Read and write	Normal
8	START2	CIQ(CIO)	Decimal		0	7	1 Bool	Read and write	Normal
9	IND_FWD1	CIQ(CIO)	Decimal		100	0	1 Bool	Read and write	Normal
10	IND_RVS1	CIQ(CIO)	Decimal		100	1	1 Bool	Read and write	Normal
11	IND_FWD2	CIQ(CIO)	Decimal		100	2	1 Bool	Read and write	Normal
12	IND_RVS2	CIQ(CIO)	Decimal		100	3	1 Bool	Read and write	Normal
13	IND_STOP1	CIQ(CIO)	Decimal		100	6	1 Bool	Read and write	Normal
14	IND_STOP2	CIQ(CIO)	Decimal		100	7	1 Bool	Read and write	Normal
15	READY1	CIQ(CIO)	Decimal		100	4	1 Bool	Read and write	Normal
16	READY2	CIQ(CIO)	Decimal		100	5	1 Bool	Read and write	Normal
17	STOP_RUN2	W(W)	Decimal		0	10	1 Bool	Read and write	Normal
18	STOP_RUN1	W(W)	Decimal		0	11	1 Bool	Read and write	Normal
19	OVERLOAD1	CIQ(CIO)	Decimal		0	8	1 Bool	Read and write	Normal
20	OVERLOAD2	CIQ(CIO)	Decimal		0	10	1 Bool	Read and write	Normal

Gambar 4. 7 Address Register PLC

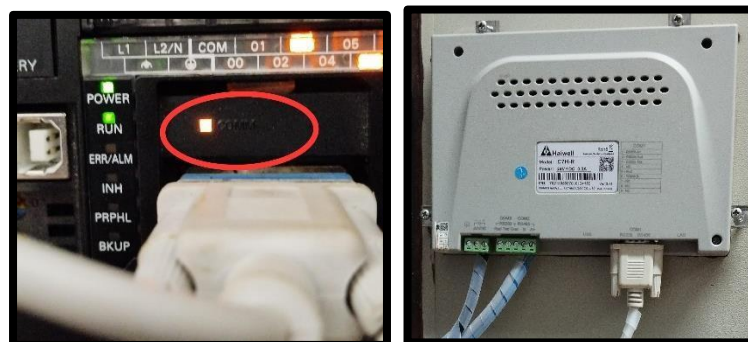
Variable name	Register type	Address format	Register address	Bit address	Address length	Data type	The mode of reading and writing
1 RUN_HZ	4x(Analog output,Function code 03 06)	Decimal	3202			1 Integer	Read and write
2 SET_HZ	4x(Analog output,Function code 03 06)	Decimal	8502			1 Integer	Read and write
3 RPM	4x(Analog output,Function code 03 06)	Decimal	8604			1 Integer	Read and write
4 CMD	4x(Analog output,Function code 03 06)	Decimal	8501			1 Integer	Read and write
5 MOTOR_VOLTAGE	4x(Analog output,Function code 03 06)	Decimal	3207			1 Integer	Read and write
6 MOTOR_CURENT	4x(Analog output,Function code 03 06)	Decimal	3204			1 Integer	Read and write
7 CMD_SPEED	4x(Analog output,Function code 03 06)	Decimal	8601			1 Integer	Read and write
8 SET_RPM	4x(Analog output,Function code 03 06)	Decimal	8602			1 Integer	Read and write
9 PID_ON	4x(Analog output,Function code 03 06)	Decimal	11908	0		1 Bool	Read and write
10 Ki	4x(Analog output,Function code 03 06)	Decimal	11941			1 Integer	Read and write
11 Kp	4x(Analog output,Function code 03 06)	Decimal	11942			1 Integer	Read and write
12 Kd	4x(Analog output,Function code 03 06)	Decimal	11943			1 Integer	Read and write
13 ipi	4x(Analog output,Function code 03 06)	Decimal	11920			1 Integer	Read and write
14 acc	4x(Analog output,Function code 03 06)	Decimal	9001			1 Integer	Read and write
15 dcc	4x(Analog output,Function code 03 06)	Decimal	9002			1 Integer	Read and write

Gambar 4. 8 List Register Address Inverter

Pada *device inverter*, *Register Address* sesuai dengan yang terdapat pada *modbus Address Register inverter ATV12* dan dapat dilihat pada *user manual inverter ATV12*. Daftar *Address Register* yang digunakan pada alat ini dapat dilihat pada Gambar 4.8 di atas.

4.1.3 Perakitan Sistem Komunikasi PLC Dengan HMI.

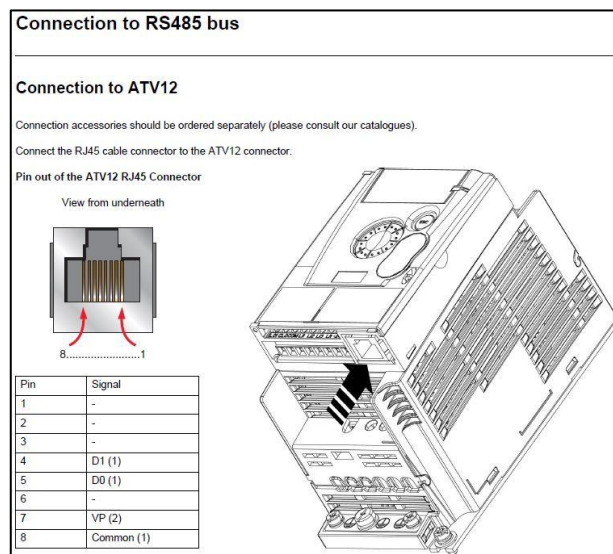
Untuk menghubungkan antara PLC dengan HMI bisa dengan menggunakan beberapa sistem komunikasi, diantaranya dengan menggunakan *serial RS232*, *private network (LANinternet)*, dan *wireless communication system*. Untuk percobaan kali ini digunakan *serial komunikasi RS232* untuk menghubungkan antara PLC Omron CP1E dengan HMI. Berikut tampilan dari *wiring komunikasi PLC dengan HMI menggunakan kabel RS 232* pada Gambar 4. 9 di bawah ini.



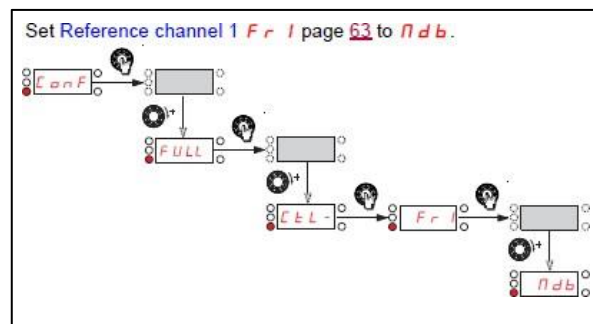
Gambar 4. 9 Tampilan Koneksi RS 232

4.1.4 Perakitan Sistem Komunikasi HMI Dengan *Inverter* .

Komunikasi antara HMI dengan *inverter* dengan *modbus protocol* dapat menggunakan *port* RJ45 yang dimodifikasi menjadi RS485 seperti yang ditunjukkan pada Gambar 4.10. Komunikasi ini berfungsi 2 arah dimana HMI memberikan *setpoint* kecepatan motor, Parameter PID dan parameter akselerasi dekelerasi. *Inverter* mengirimkan data *monitoring* berupa data kecepatan motor, frekuensi, tegangan dan arus.



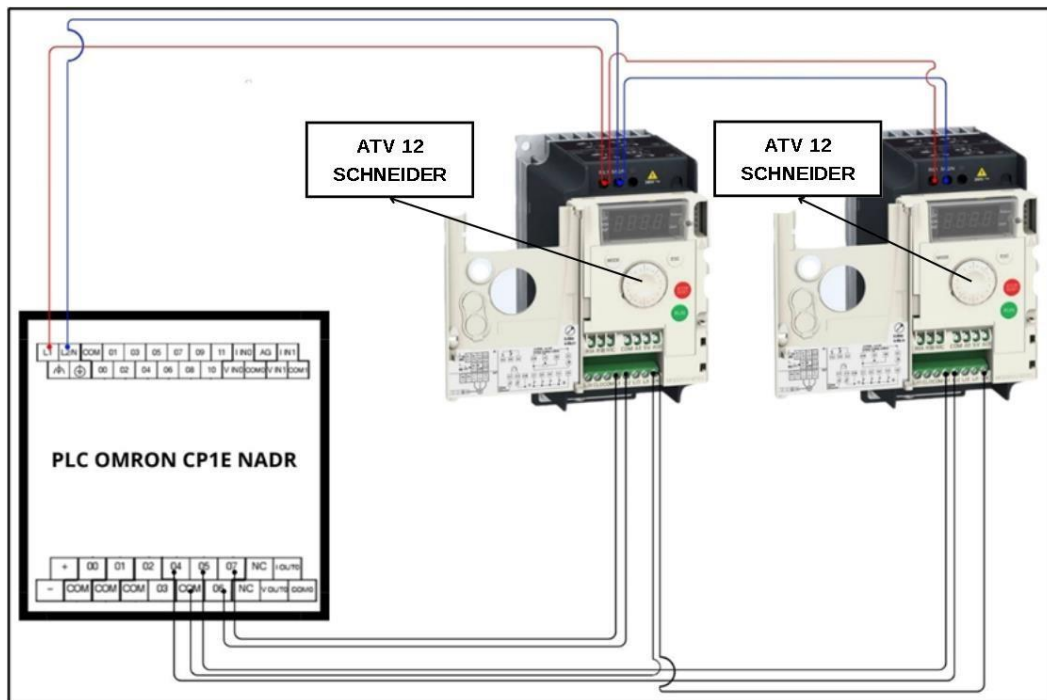
Gambar 4. 10 Pin Konfigurasi RJ45 pada *Inverter* ATV12



Gambar 4. 11 Set Reference Channel

Sebelum menghubungkan kedua *port*, harus dilakukan konfigurasi pada *inverter* untuk mengaktifkan protokol *modbus communication* pada internal *inverter* dengan cara yang ditunjukkan pada Gambar 4. 11 di atas. Tekan dial pada *inverter* , kemudian putar hingga *display* memberi tampilan *ConF => FULL => CtL => Fr I => ndb*.

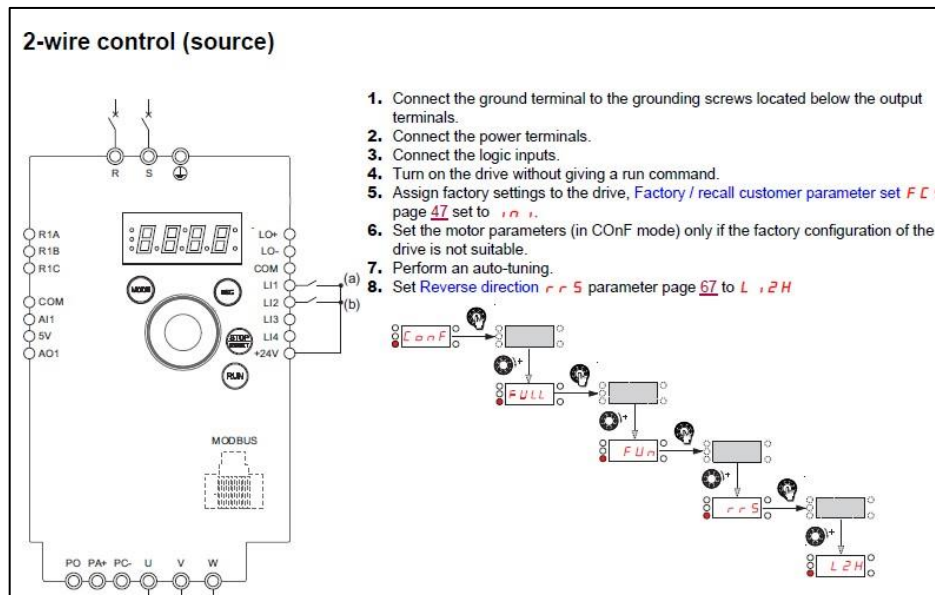
4.1.5 Pemasangan Wiring Sistem Komunikasi PLC Dengan Inverter .



Gambar 4. 12 Wiring PLC Dengan Inverter

Komunikasi PLC dengan *Inverter* memiliki fungsi untuk memberi perintah *Forward Reverse* dan *Stop*. Keluaran PLC dengan alamat 100.04 dihubungkan dengan L1 *inverter 1* untuk memberi perintah *forward*, Keluaran 100.5 PLC dihubungkan dengan L2 *inverter 1* untuk memberi perintah *reverse*, , Keluaran 100.6 PLC dihubungkan dengan L1 *inverte2* untuk memberi perintah *forward* motor 2, Keluaran 100.7 PLC dihubungkan dengan L2 *inverter 2* untuk memberi perintah *reverse* motor 2 dan pin 24V pada kedua *inverter* dihubungkan dengan com PLC untuk pin 100.04 – 100.07. Kemudian hubungkan juga pin *supply* tegangan AC PLC dan *inverter* dengan sumber tegangan 220 ACV. Tampilan *Wiring diagram* PLC dengan *inverter* dapat dilihat pada Gambar 4. 12.

Setelah menghubungkan *wiring* PLC dengan *inverter* perlu juga dilakukan konfigurasi pada *inverter* agar dapat menjalankan perintah *Forward Reverse* melalui konfigurasi *2 wire*. Langkah – langkah konfigurasi ini dapat dilihat pada *user manual* ATV12 seperti yang terlihat pada Gambar 4. 13 pada halaman selanjutnya.



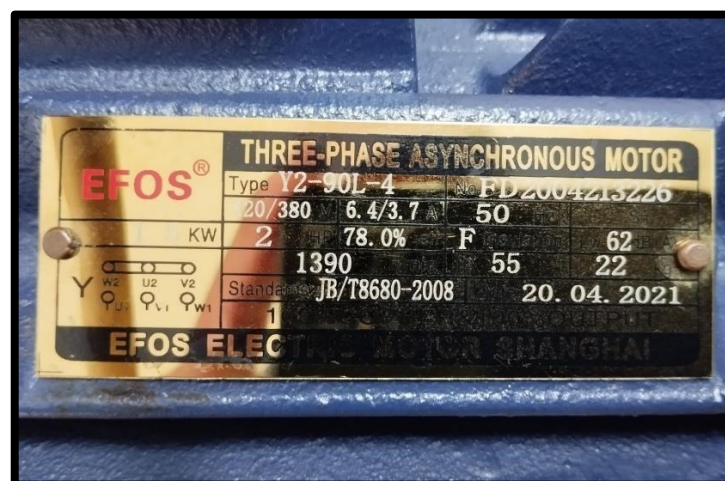
Gambar 4. 13 Two Wire Control Configuration

4.1.6 Pemasangan Wiring Inverter Dengan Motor.

Spesifikasi pada kedua motor motor induksi 3 phase yang yang digunakan pada penelitian kali ini dapat dilihat pada Tabel 4.2 dan Gambar 4. 14 di bawah ini.

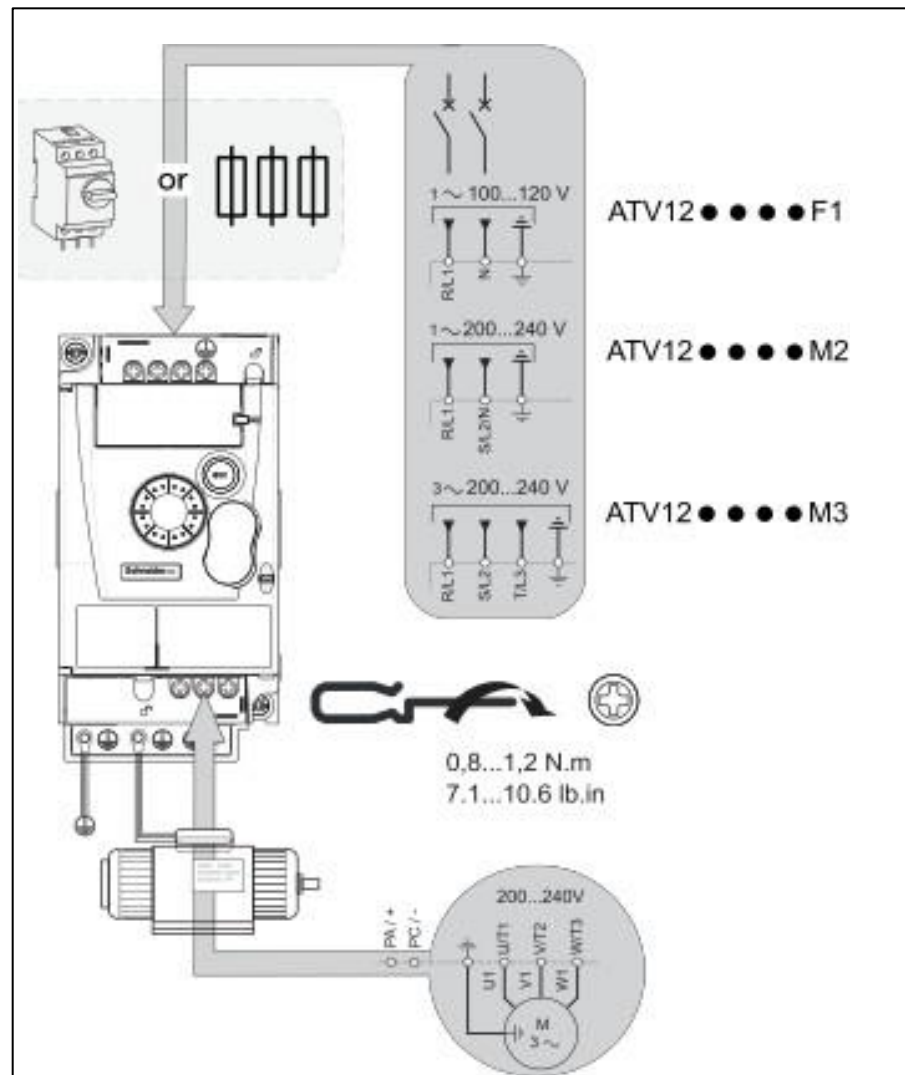
Tabel 4. 2 Spesifikasi Motor

Motor 1	Motor 2
Daya : 1,5 kW	Daya : 1,5 kW
Kecepatan Maksimal : 1500 Rpm	Kecepatan Maksimal : 1500 Rpm
Tegangan : 220 / 380 Volt	Tegangan : 220 / 380 Volt
Frekuensi : 50 Hz	Frekuensi : 50 Hz
Arus : 6.4 / 3.7 A	Arus : 6.4 / 3.7 A
<i>Pole Pair</i> : 4	<i>Pole Pair</i> : 4



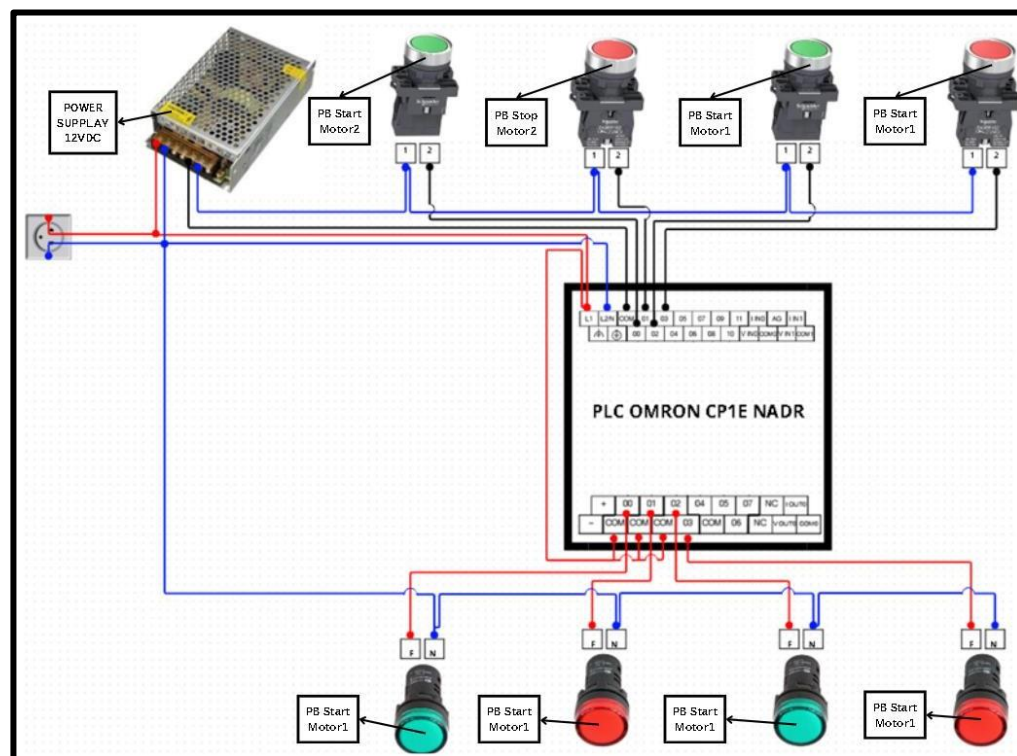
Gambar 4. 14 Name Plate Motor

Jenis *inverter* yang digunakan adalah Altivar 12 yang diperuntukkan untuk aplikasi pada pompa, conveyor, dan juga fan dalam dunia industri. Batas maksimum dari Altivar 12 adalah sebesar 2.5 kw yang maka dari itu *inverter* ini dipergunakan untuk mengatur kecepatan jenis motor yang ringan. Gambar 4. 15 di bawah merupakan *Wiring inverter* dengan motor induksi 3 fasa.



Gambar 4. 15 *Wiring Inverter* Dengan Motor

4.1.7 Wiring Push button dan Lampu Indikator Dengan PLC



Gambar 4. 16 Wriung Diagram PLC Dengan *Push button* Dan Lampu Indikator

Gambar 4. 16 Menjelaskan tentang koneksi antara PLC dengan *push button* dan lampu indikator yang berfungsi sebagai pengendali operasi dari kontaktor.

4.1.8 Perakitan Sistem SCADA

Perancangan sistem SCADA pada penelitian ini menggunakan HMI yang memiliki integrasi IoT, namun HMI masih membutuhkan *router* eksternal untuk mendapatkan sinyal internet. Pada Gambar 4. 17 dan Gambar 4. 18 menunjukkan koneksi dari *router* dengan sumber internet dan *router* dengan HMI.



Gambar 4. 17 Port Sumber Internet dan keluaran 1 Router



Gambar 4. 18 Port LAN Pada Bagian Belakang HMI



Gambar 4. 19 Network Setting HMI Haiwell

Untuk menghubungkan HMI dengan perangkat SCADA dapat dilakukan dengan memindai *barcode* yang berada pada menu *network* HMI seperti yang ditunjukkan Gambar 4. 19. Proses pindai *barcode* ini dapat dilakukan menggunakan perangkat *smartphone* melalui aplikasi haiwel SCADA *cloud* atau melalui web haiwel. Tampilan sistem SCADA dapat dilihat pada Gambar 4. 20 di bawah ini.



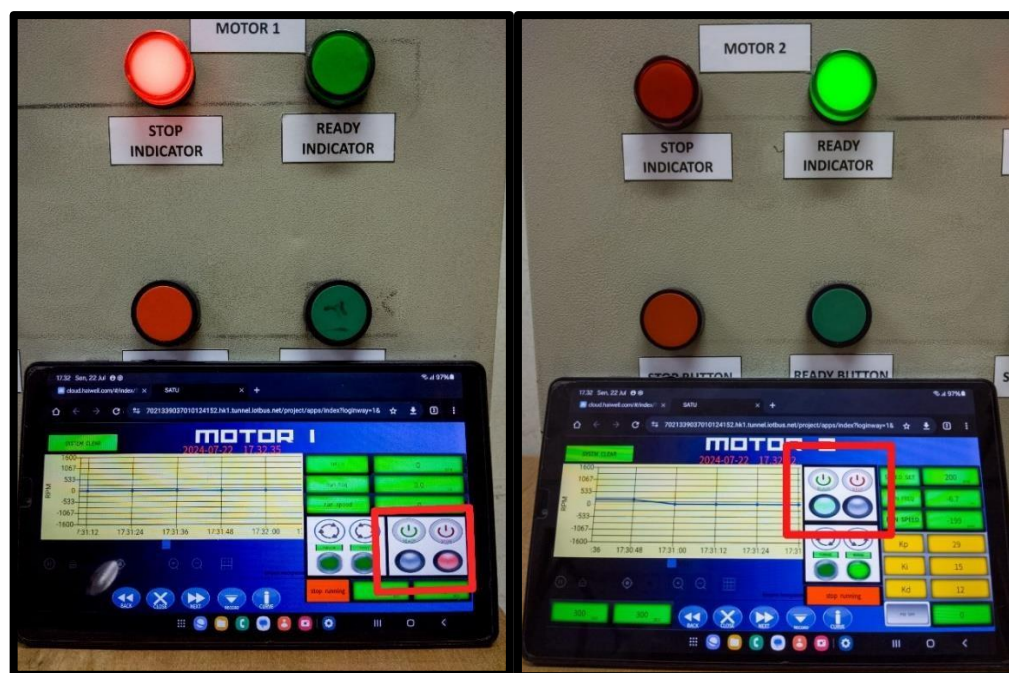
Gambar 4. 20 Tampilan Sistem SCADA Melalui Smartphone dan Laptop

4.2 Pengujian Alat

Pengujian dilakukan mulai dari keselarasan *Push button* yang terdapat pada *Hardware* dan HMI, fungsi *Forward Reverse*, pengatur kecepatan dan *respon* motor terhadap perubahan parameter PID.

4.2.1 Pengujian *Push button* Dan Lampu Indikator

Pengujian dilakukan dengan menekan *Push button* pada *Hardware* dan *SCADA* secara bergantian. Ketika *Push button ready* ditekan maka lampu hijau akan menyala dan lampu merah mati, selain itu juga menghubungkan sumber listrik motor. Kemudian ketika *Push button stop* ditekan maka lampu merah akan menyala dan lampu hijau mati untuk menandakan kontaktor tidak aktif dan sumber listrik motor terputus. Hasil pengujian dapat dilihat pada Gambar 4. 21.



Gambar 4. 21 *Push button* Dan Lampu Indikator Pada *Hardware* dan *SCADA*

4.2.2 Pengujian Pengatur Kecepatan

Pengujian dilakukan dengan memasukan *setpoint* kecepatan motor listrik melalui *interface SCADA* seperti yang ditunjukkan oleh Gambar 4. 22. Kemudian data yang ditampilkan pada *interface* akan dibandingkan dengan data yang sebenarnya terukur melalui *taco* meter. Perbandingan data tersebut dapat dilihat

pada Tabel 4.3. Dari perbandingan data tersebut dapat ditentukan nilai *error* dengan menggunakan Persamaan 4 yang telah dimasukkan ke kolom tabel *microsoft excel*.

Pada Tabel 4.3 dapat dilihat bahwa semakin besar *setpoint* yang dimasukkan, maka akan semakin besar presentase *error* yang terjadi.

$$Error = \frac{(Data\ Interface - Data\ Sebenarnya)}{Data\ Sebenarnya} \times 100 \dots \text{Persamaan 4}$$

Dapat dibuktikan menurut Persamaan 1, mencari nilai frekuensi pada data *interface* (RPM) didapat hasil sebagai berikut :

i. Diketahui : $N_s = 50$;

$$p = 4;$$

Ditanyakan : $f = \dots?$

Jawab :

$$N_s = \frac{120 \times f}{p} \dots \text{Persamaan 1}$$

$$50 = \frac{120 \times f}{4}$$

$$50 \times 4 = 120 \times f$$

$$f = \frac{200}{120}$$

$$f = 1,66 \text{ Hz}$$

Dapat dibuktikan menurut Persamaan 1, mencari nilai frekuensi pada data sebenarnya (RPM) didapat hasil sebagai berikut :

ii. Diketahui : $N_s = 71$;

$$p = 4;$$

Ditanyakan : $f = \dots?$

Jawab :

$$71 = \frac{120 \times f}{p} \dots \text{Persamaan 1}$$

$$71 = \frac{120 \times f}{4}$$

$$71 \times 4 = 120 \times f$$

$$f = \frac{284}{120}$$

$$f = 2,36 \text{ Hz}$$



Gambar 4. 22 Pengukuran Kecepatan Dengan Taco Meter

Tabel 4. 3 Perbandingan Data pengukuran kecepatan Dengan Data Sebenarnya

NO	DI (RPM)	DS (RPM)	DI (f)	DS (f)	$ERROR f$ (%)	$ERROR RPM$ (%)
1	50	71	1,66	2,36	42	-29,66101695
2	100	120	3,33	4	20	-16,75
3	550	554	18,3	18,46	0,72727272	-0,704225352
4	800	802	26,7	26,73	0,25	-0,26187804
5	1000	1002	33,3	33,4	0,2	-0,209580838

Keterangan :

DI = Data *Interface*

DS = Data Sebenarnya

4.2.3 Pengujian Parameter PID

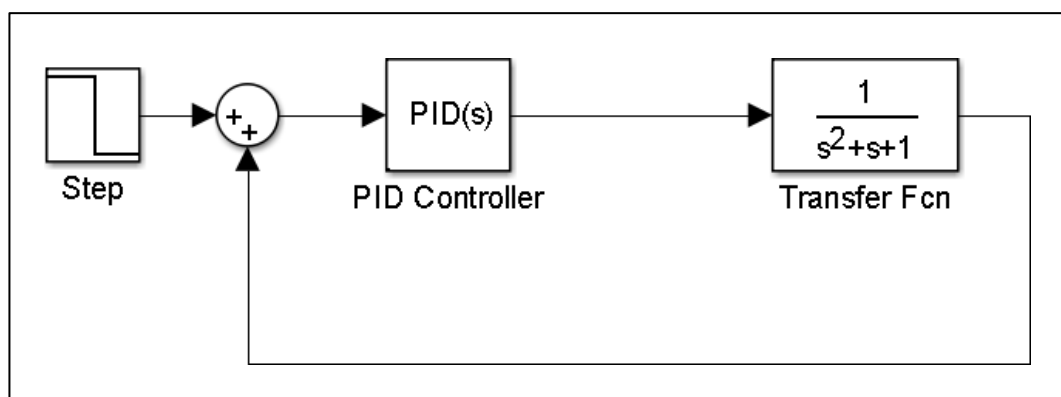
Controller parameters	
	Tuned
Kp	26.7937
Ki	14.6511
Kd	12.25
Tf	
Performance and robustness	
	Tuned
Rise time	5.66 seconds
Settling time	10.9 seconds
Overshoot	0%
Peak	0.998
Gain margin	Inf dB @ NaN rad/s
Phase margin	84 deg @ 16.6 rad/s
Closed-loop stability	Stable

Gambar 4. 23 Tuning Parameter PID Melalui Matlab

Proses *Tuning* parameter PID dilakukan dengan metode *trial & gain* yang dilakukan secara simulasi melalui *software* matlab. Sehingga didapatkan parameter PID yang ideal sebagai berikut. $K_P = 26$ $K_i = 14$ dan $K_d = 12.$, seperti yang ditunjukkan pada Gambar 4. 23.

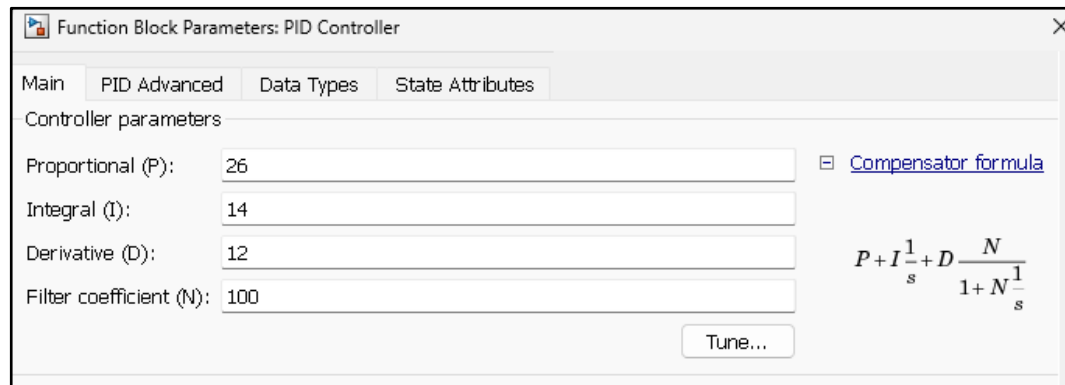
4.2.3.1 Pengujian Parameter PID Menggunakan *software* Matlab

Pengujian dilakukan dengan memasukan *function block parameters* seperti yang ditunjukkan pada Gambar 4. 24 di bawah ini.

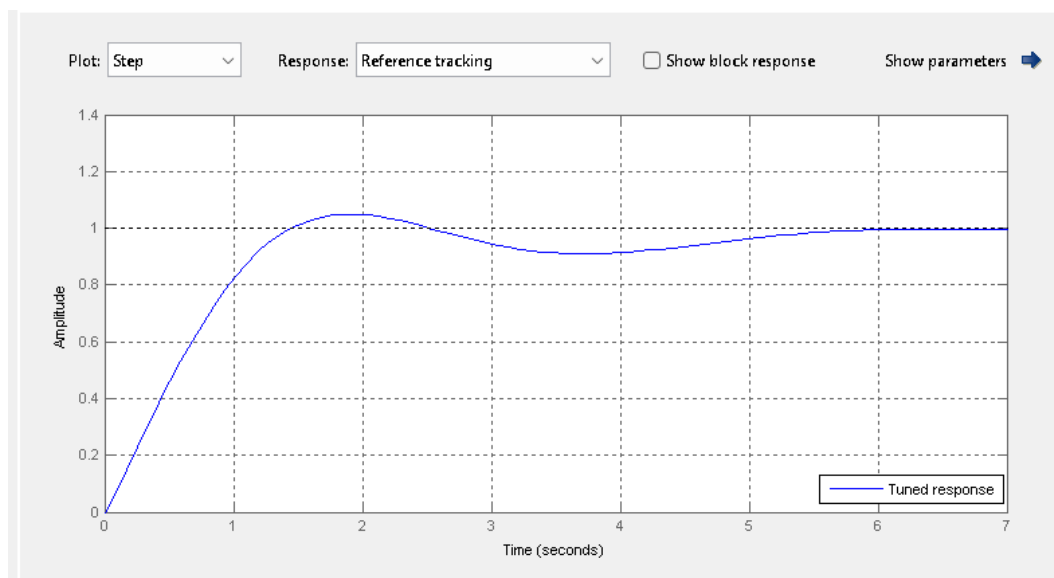


Gambar 4. 24 Function Block Parameters

Setelah memasukan *Function Block Parameters* selanjutnya masukan parameter PID sebesar $K_p = 26$ $K_i = 14$ $K_d = 12$ seperti yang ditunjukkan pada Gambar 4. 25 di bawah ini.



Gambar 4. 25 Setting Parameter PID



Gambar 4. 26 Hasil Pengujian Dengan $K_p = 26$ $K_i = 14$ $K_d = 12$

Gambar 4.26 menunjukkan hasil dari pengujian menggunakan *software* matlab, dapat diperhatikan pada pengujian dengan parameter tersebut masih terdapat osilasi yang berarti kecepatan yang dihasilkan masih tidak stabil.

4.3.2.2 Pengujian Parameter PID Secara Langsung



Gambar 4. 27 Setting Parameter PID Pada SCADA

Pengujian dilakukan dengan memberikan parameter PID yang telah diperoleh melalui simulasi kemudian di masukan kedalam alat melalui sistem SCADA seperti yang ditunjukkan pada Gambar 4. 27. Kemudian motor 2 dijalankan dengan keadaan berbeban. Beban putaran motor pada penelitian kali ini menggunakan *pulley* perbandingan sebesar 5 : 1, dan motor 1 dijalankan dengan *set point* sebesar 200 Rpm tanpa beban dan PID.

Hasil dari pengaturan parameter tersebut kemudian dapat dilihat melalui data *historical* grafik maupun tabel yang telah ditambahkan pada tampilan HMI. Kemudian data ini dapat diunduh melalui sistem SCADA. Data tersebut dapat dilihat pada Gambar 4. 28 dan Tabel 4. seperti yang di bawah ini.

$$K_p = 26, K_i = 14, K_d = 12.$$



Gambar 4. 28 *Curva Respon* Kecepatan Motor Terhadap Waktu

Tabel 4. 4 Data Respon $K_p = 26$, $K_i = 14$, $K_d = 12$

TIME	MOTOR 1		MOTOR 2	
	FREQUENCY	SPEED	FREQUENCY	SPEED
23:55:34	0	0	0	0
23:55:35	-0.9	-28	0	0
23:55:36	-2.9	-91	0.5	7
23:55:37	-4.8	-145	2.1	63
23:55:38	-6.6	-198	3.9	118
23:55:39	-6.7	-200	5.6	168
23:55:40	-6.7	-199	6.9	209
23:55:41	-6.7	-200	6.6	198
23:55:42	-6.7	-199	6.6	197
23:55:43	-6.7	-199	7	208
23:55:44	-6.7	-200	7	208
23:55:45	-6.7	-200	6.6	198
23:55:46	-6.7	-200	6.6	197
23:55:47	-6.7	-200	6.6	197
23:55:48	-6.7	-200	6.7	202
23:55:49	-6.7	-200	6.9	206
23:55:50	-6.7	-200	6.7	199
23:55:51	-6.7	-200	6.6	197
23:55:52	-6.7	-200	6.6	197
23:55:53	-6.7	-200	6.6	197
23:55:54	-6.7	-200	7	208
23:55:55	-6.7	-200	7	208
23:55:56	-6.7	-200	6.6	197
23:55:57	-6.7	-200	6.6	196
23:55:58	-6.7	-200	6.6	197
23:55:59	-6.7	-200	6.9	206
23:56:00	-6.7	-200	6.9	206

Dari hasil percobaan diatas dapat dilihat *Curva* pada Gambar 4. 25 masih terjadi *overshoot* sebesar 520 rpm. Maka dari itu dilakukan penentuan ulang parameter PID dengan metode *trial and eror* dengan menaikkan nilai parameter hingga mendapat hasil yang ideal. Hingga ditemukan parameter yang ideal menjadi sebagai berikut, $K_p = 29$, $K_i = 15$ dan $K_d = 12$. Sehingga mendapat hasil seperti yang ditunjukkan *Curva* pada Gambar 4. 29 dan Tabel 4.4 di bawah ini.



Gambar 4. 29 *Curva Respon* $K_p = 29$, $K_i = 15$ dan $K_d = 12$.

Tabel 4. 5 *Data Respon* $K_p = 29$, $K_i = 15$, $K_d = 12$

$K_p = 29, K_i = 15, K_d = 12$				
TIME	MOTOR 1		MOTOR 2	
	FREQUENCY	SPEED	FREQUENCY	SPEED
00:21:20	0	0	0	0
00:21:21	-0.2	-4	0	0
00:21:22	-1.9	-55	0	0
00:21:23	-3.7	-108	0	0
00:21:24	-5.3	-139	0.1	5
00:21:25	-6.4	-193	1.1	27
00:21:26	-6.7	-200	1.9	57
00:21:27	-6.7	-200	2.8	83
00:21:28	-6.7	-200	3.6	107
00:21:29	-6.7	-199	4.5	130
00:21:30	-6.7	-200	5.2	155
00:21:31	-6.7	-199	6.1	184
00:21:32	-6.7	-200	6.7	199
00:21:33	-6.7	-200	6.7	200
00:21:34	-6.7	-200	6.7	200
00:21:35	-6.7	-200	6.7	200
00:21:36	-6.7	-199	6.7	199
00:21:37	-6.7	-200	6.7	200
00:21:38	-6.7	-200	6.7	200
00:21:39	-6.7	-200	6.7	200

Dari percobaan di atas dapat dilihat sudah tidak terjadi lagi *overshoot* dan osilasi. Sehingga dapat dikatakan terdapat selisih sebesar 2% antara *Tuning* parameter yang dilakukan melalui simulasi matlab dengan data yang sebenarnya terjadi pada implementasi alat.

Dilakukan pengujian ulang untuk memastikan bahwa sistem PID benar – benar bekerja dengan menaikkan nilai K_p K_i dan K_d secara bergantian dengan kondisi tanpa beban sehingga mendapatkan hasil *Curva* seperti yang ditunjukkan pada Gambar 4. 30 dan Gambar 4. 31.

4.2.3.3 Pengujian Parameter *Propotional*

Pengujian ini dilakukan dengan hanya memberi parameter *propotional* ($K_p = 29$, $K_i = 1$ $K_d = 0$). Untuk K_i dan K_d diberi 1 dan 0. Sehingga memberi hasil seperti yang ditampilkan pada Gambar 4. 27. Dapat diperhatikan bahwa kecepatan masih terjadi osilasi dan *overshoot* pada saat *risetime*. Sehingga masih tidak bisa dikatakan stabil.



Gambar 4. 30 Hasil Pengujian Dengan $K_p = 29$ $K_i = 1$ $K_d = 0$

4.2.3.4 Pengujian Parameter *Integral*



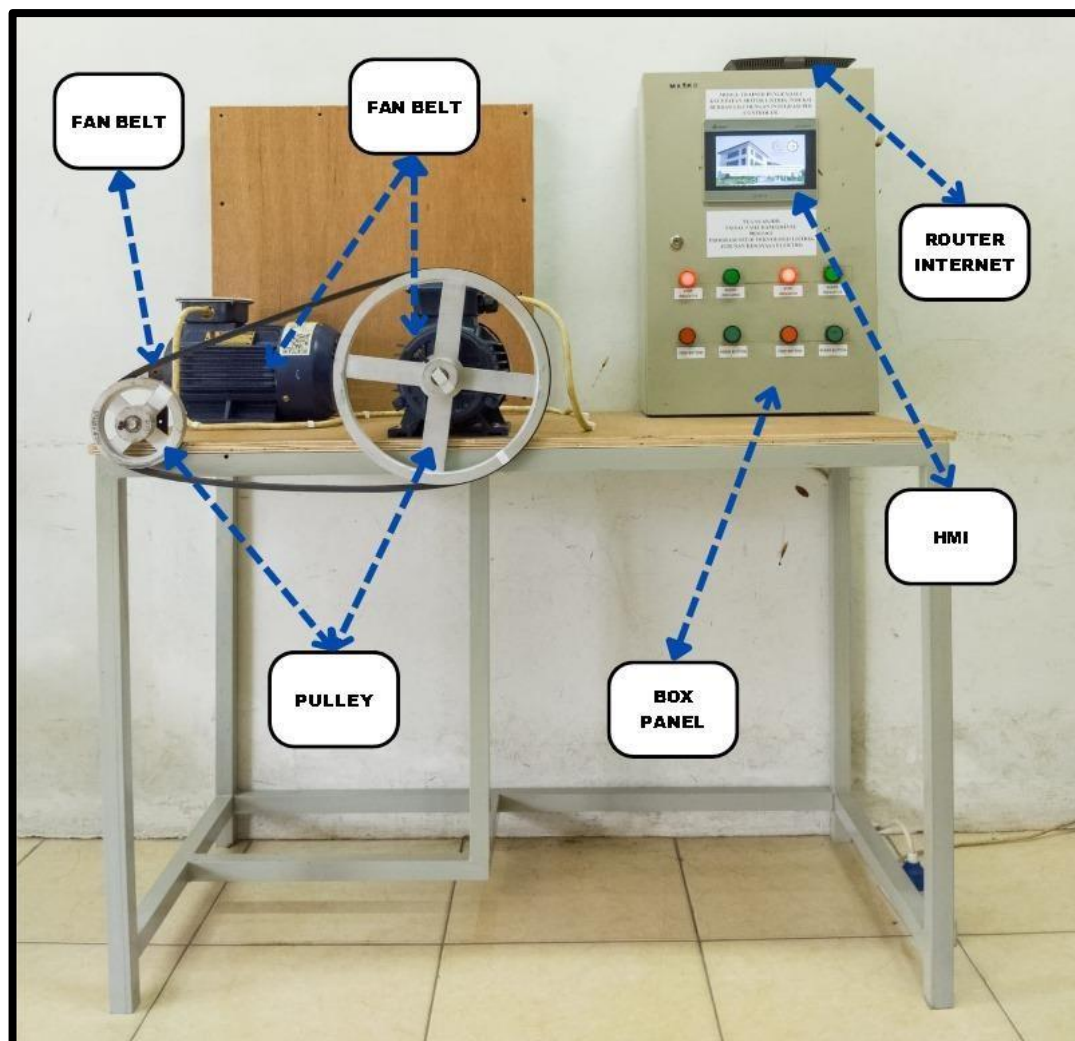
Gambar 4. 31 Hasil Pengujian Dengan $K_i = 29$ $K_p = 15$ $K_d = 0$

Pada Gambar 4. 31 dapat dilihat efek dari parameter *integral*. Ketika diberi parameter $K_i = 29$ $K_d = 0$, *overshoot* yang terjadi menjadi lebih sedikit namun *overshoot* yang terjadi semakin tinggi. Dapat dikatakan bahwa parameter *integral* dapat mengurangi *steady state*.

Dari beberapa percobaan di atas dapat membuktikan bahwa parameter PID dapat mempengaruhi respon kecepatan motor listrik. Dan jika parameter PID yang dimasukan telah sesuai maka akan memberikan respon motor listrik yang stabil.

4.3 Tampilan Akhir Alat

Setelah melakukan beberapa tahapan perancangan maka tampilan hasil dari perancangan alat ini dapat dilihat pada Gambar 4. 32 berikut.



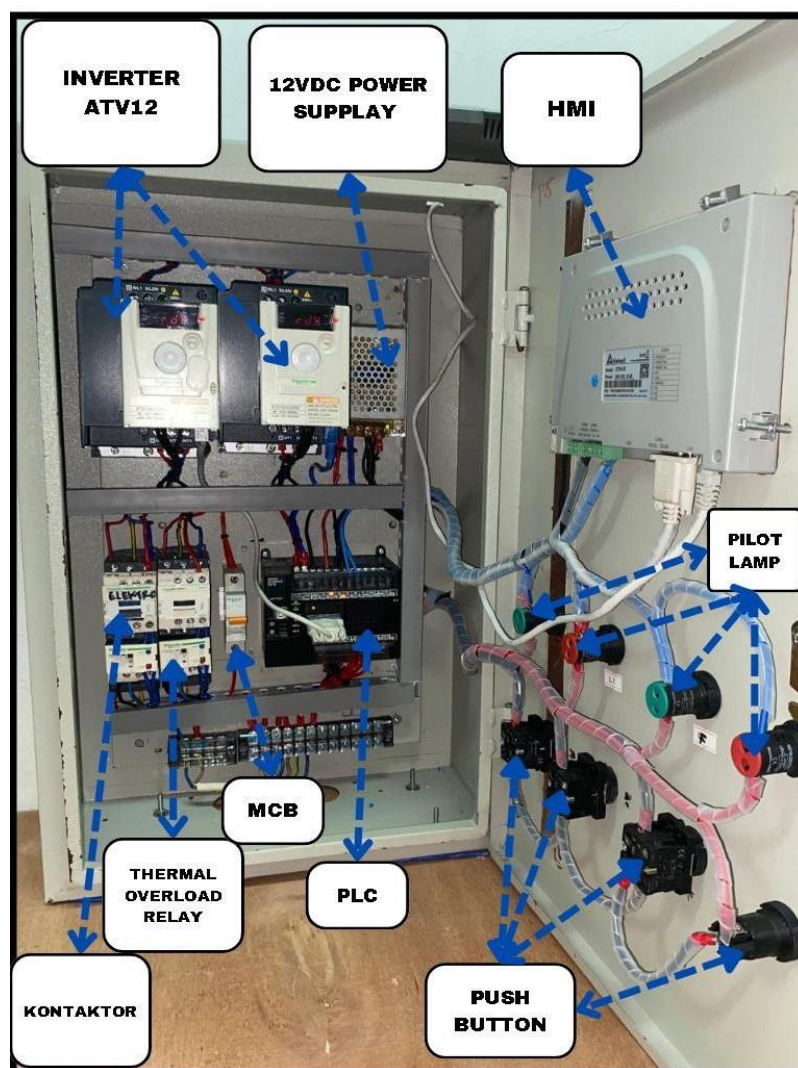
Gambar 4. 32 Tampilan Akhir Alat

4.4 Tampilan Akhir Panel

Pada bagian panel, terdapat komponen – komponen utama yang menunjang kinerja dari alat ini. Beberapa komponen tersebut secara detail dapat dilihat sebagai berikut.

4.4.1 Tampilan Dalam Panel

Peletakan komponen – komponen berikut telah direncanakan dengan sedemikian rupa sehingga terpasang seperti yang ditampilkan pada Gambar 4. 33 di berikut.



Gambar 4. 33 Tampilan Dalam Panel

4.4.2 Tampilan Luar Panel

Bagian luar panel merupakan yang menjadi bagian antar muka dari alat dengan pengguna alat. Selain terdapat beberapa tombol dan indikator, juga terdapat HMI haiwell seperti yang ditunjukk oleh Gambar 4. 34 berikut.



Gambar 4. 34 Tampilan Luar Panel