



PEMROGRAMAN PLC SIEMENS UNTUK APLIKASI SISTEM MIXING PLANT BERBASIS SCADA

Dr. Murie Dwiyaniti, S.T., M.T. ¹,
Silowardono, S.T., M.Si. ², Rain Shaqr Dharma Setya³
^{1,2,3}Politeknik Negeri Jakarta, Jurusan Teknik Elektro, Prodi Teknik Listrik,
Jl.Prof. Dr. GA Siwabessy, Kampus UI, Depok 16425
e-mail : rain.shaqr@gmail.com

ABSTRACT

The mixing plant serves as a tool for blending raw materials into finished products. In the manufacturing industry, this system operates continuously with various raw material products, each requiring different treatments. However, in conventional mixing systems, customization of setpoint parameters cannot be freely adjusted due to limitations in component specifications. This study aims to develop an efficient mixing plant control system with customizable setpoints tailored to user requirements, implemented using a Siemens Programmable Logic Controller (PLC) control system, and applied on SCADA WinCC and Weintek HMI devices. The research employs a descriptive quantitative method. The mixing program is analyzed line by line, and testing is conducted by measuring the analog output current and motor rotation speed. The results indicate a speed reading error of 5.55% when compared to a tachometer, and an analog output current reading error of 12.39% when measured against a multimeter on the SCADA device.

Keywords : Automation, Industrial Automation, Mixing Plant, Process Control, Programmable Logic Controller (PLC), Supervisory Control And Data Acquisition (SCADA)

ABSTRAK

Mixing plant digunakan sebagai alat pencampur bahan mentah menjadi bahan jadi. Pada industry manufaktur, sistem ini digunakan secara terus menerus dengan berbagai produk bahan mentah dengan perlakuan yang berbeda-beda. Namun pada sistem mixing konvensional, kustomisasi setpoint parameter pada sistem mixing tidak dapat diatur secara bebas karena keterbatasan pada spesifikasi komponen. Tujuan dari penelitian ini adalah pembuatan sistem kontrol mixing plant yang efisien dengan kustomisasi setpoint yang bisa diatur sesuai dengan pengguna yang berjalan pada sistem kontrol Programmable Logic Controller (PLC) Siemens yang diaplikasikan pada perangkat SCADA WinCC dan HMI Weintek. Penelitian ini menggunakan metode deskriptif kuantitatif. Program mixing dilakukan analisa pada setiap baris program serta pengujian dilakukan pengukuran terhadap arus analog output dan kecepatan putar motor. Hasil penelitian adalah nilai error pembacaan kecepatan motor terhadap alat ukur tachometer yang mencapai 5,55% dan nilai arus analog output pembacaan pada perangkat SCADA terhadap alat ukur multimeter yang mencapai 12,39%.

Kata kunci : Kontrol Proses, Mixing Plant, Otomasi, Otomasi Industri, Programmable Logic Control (PLC), Supervisory Control And Data Acquisition (SCADA)

PENDAHULUAN

Proses *mixing* digunakan umum pada industri yang berkaitan dengan perubahan fisik dan kimia. Secara umum proses *mixing* digunakan pada pemrosesan bahan pangan seperti makanan, obat, kertas, dan minuman. [1]. Dengan bergantinya bahan yang diproses diperlukan perlakuan yang berbeda pula antar bahan. PLC adalah komponen yang memiliki *microprocessor* yang digunakan sebagai kontrol pada proses otomatisasi yang bekerja dengan cara mengeksekusi program yang disimpan pada memori [2].

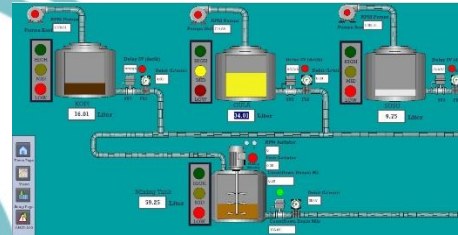
Proses *mixing* yang belum terotomasi berpotensi mempersulit pengguna dalam mengkustomisasi *setpoint* pada pengoperasian *mixing* lebih dari 1 jenis bahan mentah. Selain itu, kontrol yang otomatis mempermudah pengguna untuk melakukan akuisisi kontrol jarak jauh melalui perangkat SCADA.

Tujuan dari penelitian ini adalah untuk menciptakan kontrol PLC untuk proses *mixing plant* yang diterapkan pada aplikasi perangkat SCADA yang dijalankan melalui WinCC dan *Human Machine Interface* (HMI). Program kontrol *mixing plant* berfokus kepada kontrol kecepatan dan durasi yang dapat dikustomisasi pada perangkat SCADA. Metode pemrograman menggunakan *Function Block* (FB) dengan bahasa pemrograman *ladder*. Metode ini dapat mempersingkat *troubleshooting* pada program karena masing-masing proses dapat dikelompokkan pada 1 *Function Block* (FB).

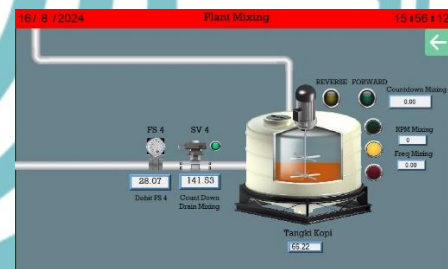
METODE PENELITIAN

Proses pengaturan kecepatan motor pada proses *mixing* dapat dilihat pada gambar 1 dan 2. Tangki *mixing* mendapatkan suplai bahan mentah dari *plant raw material*. Pengoperasian *mixing* dilakukan dengan 2 mode, yaitu *manual*

dan *auto*. Pengoperasian *manual* menggunakan potensiometer sebagai pengatur kecepatan motor *mixing*. Pengaturan arah putaran dilakukan pada perangkat SCADA. Pengoperasian *auto* menggunakan perangkat SCADA sebagai *input setpoint* kecepatan, durasi, dan arah putaran motor *mixing*.



Gambar 1. Tampilan Plant Raw Material dan Mixing pada SCADA WinCC



Gambar 2. Tampilan Tangki Mixing pada HMI

Langkah-langkah pemrograman yang dilakukan adalah :

1. Lakukan analisa terhadap diagram alir proses *mixing* yang sudah dibuat.
2. Buat pengelompokan berdasarkan proses kontrol.
3. Hasil pengelompokan direalisasikan dalam bentuk *function block*.
4. Program masing-masing *function block* dalam bahasa *ladder*.

Langkah-langkah penelitian yang dilakukan adalah:

1. Operasikan proses *mixing* dalam mode *manual* dengan memutar *selector switch*.



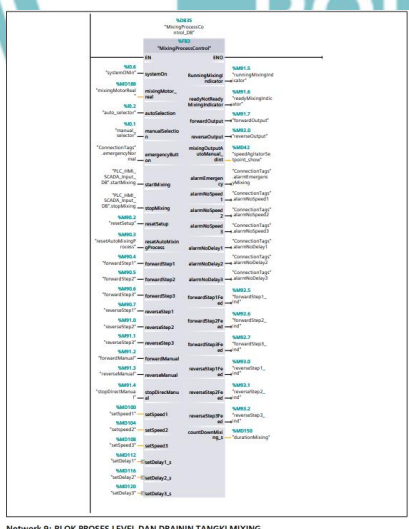
Operasikan melalui potensi dengan *interval* 0,5 V.

2. Lakukan pengukuran pada kecepatan putar motor dengan tachometer dan nilai *analog output* dengan multimeter. Catat hasil pengukuran pada tiap *interval* pengoperasian.
3. Lakukan pengamatan pada perangkat SCADA pada tampilan kecepatan motor dan nilai *analog output*. Catat pada masing-masing nilai *interval* pengoperasian.
4. Olah data pengukuran dan tampilan perangkat SCADA. Tentukan nilai hasil *error* tampilan SCADA terhadap hasil pengukuran.
5. Lakukan analisa program PLC pada proses kontrol *mixing* untuk masing-masing baris program.

HASIL DAN PEMBAHASAN

1. Analisa Program PLC Pada Proses *Mixing*

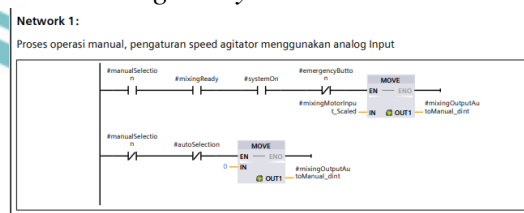
Hasil analisa program PLC pada proses *mixing* adalah berupa narasi penjelasan masing-masing baris program pada *Function Block* (FB) proses *mixing* pada gambar 3.



Gambar 3. *Function Block* (FB) Proses *Mixing*

a. Baris Program 1

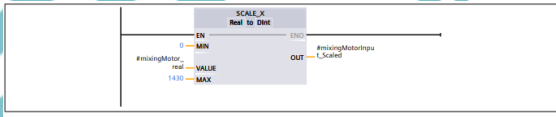
Baris program 1 pada gambar 4 berisi proses selektor untuk pengoperasian *manual* untuk pengaturan kecepatan rpm. Pada gambar 3, pengoperasian *move* data kecepatan motor akan aktif jika selektor, saklar sistem, dan indikator *mixing ready* dalam keadaan ON.



Gambar 4. Baris Program 1

b. Baris Program 2

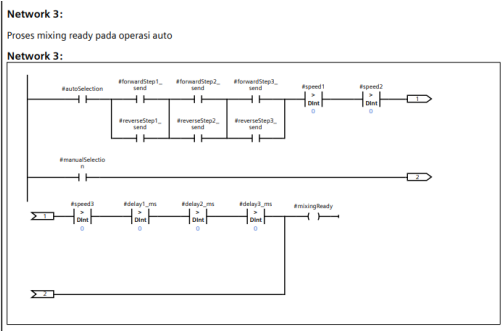
Baris program 2 berisi proses *scaling* *analog input* dari "mixingMotor_real" ke "mixingMotorInput_Scaled" yang di *scaling* dengan nilai batas bawah 0 dan batas atas 1430.



Gambar 5. Baris Program 2

c. Baris Program 3

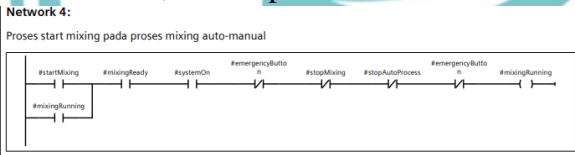
Baris program 3 berisi proses *mixing ready* untuk operasi *auto* dan *manual*. Pada proses *auto*, "mixingReady" akan TRUE apabila : Selektor *auto* = TRUE Kondisi *forward* atau *reverse step 1* = TRUE Kondisi *forward* atau *reverse step 2* = TRUE Kondisi *forward* atau *reverse step 3* = TRUE Nilai *setpoint speed 1* > 0 rpm Nilai *setpoint speed 2* > 0 rpm Nilai *setpoint speed 3* > 0 rpm Nilai *setpoint delay 1* > 0 ms Nilai *setpoint delay 2* > 0 ms Nilai *setpoint delay 3* > 0 ms Pada proses *manual*, "mixingReady" akan TRUE apabila selektor *manual* = TRUE



Gambar 6. Baris Program 3

d. Baris Program 4

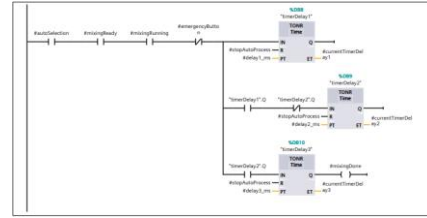
Baris program 4 berisi proses *mixing* dapat dilakukan jika “mixingReady“, systemOn, dan startMixing berstatus TRUE. Mixing dapat dihentikan dengan mengaktifkan *emergency*, tombol STOP, dan stopAutoProcess.



Gambar 7. Baris Program 4

e. Baris Program 5

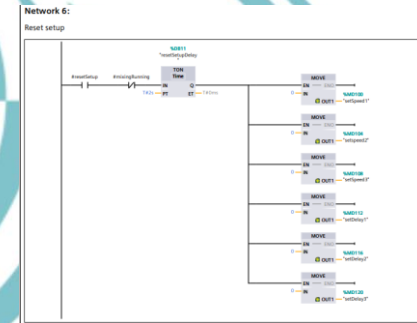
Baris program 5 berisi proses durasi *mixing* pada waktu kondisi *running* pada operasi *auto*. Proses aktif ketika selektor *auto*, “mixingReady“, dan “mixingRunning“ adalah TRUE. *Timer* menggunakan tipe *retentive* yang berarti saat *Timer In* = False, maka hitungan *timer* akan berhenti namun tidak *ter-reset*. Hitungan akan dilanjutkan kembali ketika *Timer In* = TRUE. *Timer* beroperasi berurutan dari *Timer 1*. *Timer 2* bekerja saat *timer 1 Q* = TRUE. *Timer 3* Bekerja saat *Timer 2 Q* = TRUE. *Timer 1, 2, dan 3* akan mati saat *Timer 3 Q* = TRUE.



Gambar 8. Baris Program 5

f. Baris Program 6

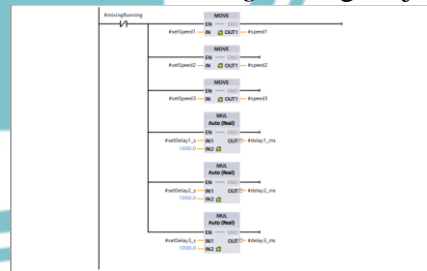
Baris program 6 berisi proses *reset setup* terhadap *speed 1, 2, 3* dan *delay 1, 2, 3*. *Reset* dilakukan dengan menekan “resetSetup“ selama 2 detik.



Gambar 9. Baris Program 6

g. Baris Program 7

Baris program 7 berisi proses *input setpoint* kecepatan dan *delay* untuk *speed 1, 2, 3* dan *delay 1, 2, 3*. Proses ini tidak bisa dioperasikan ketika *mixing* sedang berjalan.



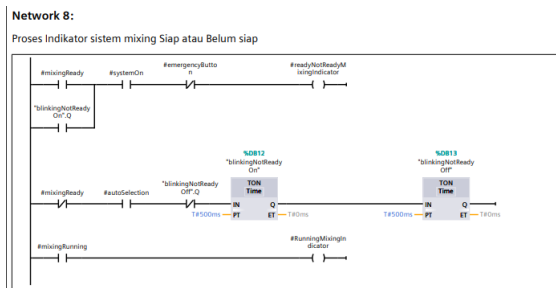
Gambar 10. Baris Program 7

h. Baris Program 8

Baris program 8 berisi proses indikator *ready mixing*. Jika *setpoint* tidak diisi pada operasi *auto*, lampu indikator “readyNotReadyMixingIndicator“ akan



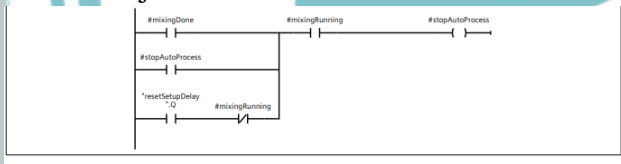
berkedip dengan *interval* 500ms. Saat *setpoint* sudah terisi, maka lampu akan menyala tanpa *interval*.



Gambar 11. Baris Program 8

i. Baris Program 9

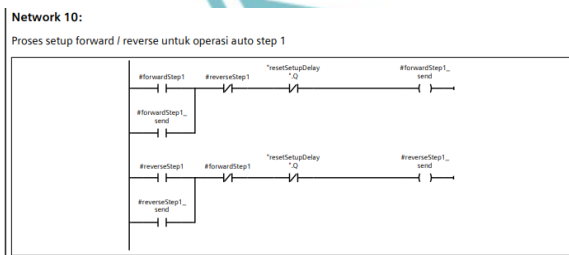
Baris program 9 berisi proses untuk menghentikan proses *mixing auto*. Proses *mixing auto* akan berhenti ketika proses *mixing* selesai atau tombol “resetSetup” ditekan saat *mixing* tidak bekerja.



Gambar 12. Baris Program 9

j. Baris Program 10

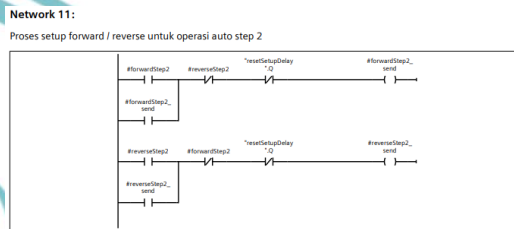
Baris program 10 berisi proses penentuan arah putaran pada *speed 1* untuk operasi *auto*. *Forward* akan aktif jika tombol *forward* ditekan. *Reverse* akan aktif jika tombol *reverse* ditekan. Untuk menonaktifkan kondisi putaran dapat dilakukan dengan menekan *resetSetup* selama 2 detik.



Gambar 13. Baris Program 10

k. Baris Program 11

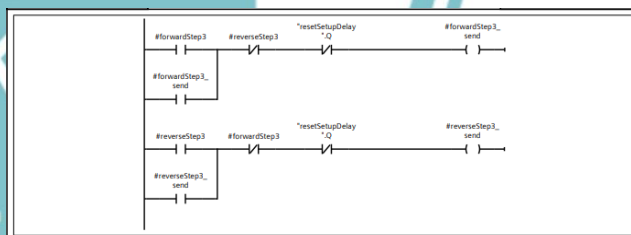
Baris program 11 berisi proses penentuan arah putaran pada *speed 2* untuk operasi *auto*. *Forward* akan aktif jika tombol *forward* ditekan. *Reverse* akan aktif jika tombol *reverse* ditekan. Untuk menonaktifkan kondisi putaran dapat dilakukan dengan menekan *resetSetup* selama 2 detik.



Gambar 14. Baris Program 11

l. Baris Program 12

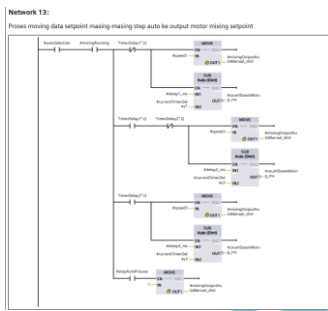
Baris program 12 berisi proses penentuan arah putaran pada *speed 3* untuk operasi *auto*. *Forward* akan aktif jika tombol *forward* ditekan. *Reverse* akan aktif jika tombol *reverse* ditekan. Untuk menonaktifkan kondisi putaran dapat dilakukan dengan menekan *resetSetup* selama 2 detik.



Gambar 15. Baris Program 12

m. Baris Program 13

Baris program 13 berisi proses *moving data setpoint* ke *output speed motor*. Proses *moving* diaktifkan melalui *timer* untuk masing-masing durasi. Selain itu, ada proses hitung mundur untuk proses *mixing auto*. Proses dilakukan dengan rumus *subtract* dari data *setpoint delay timer* dan *data value timer*.



Gambar 16. Baris Program 13

n. Baris Program 14

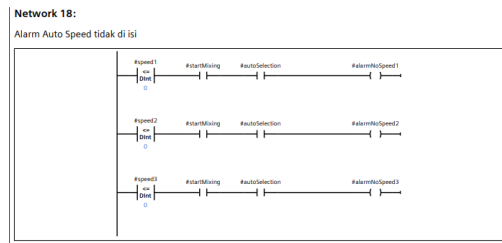
Baris program 14 berisi proses untuk menyalakan *output forward* dan *reverse* untuk pengoperasian *auto* dan *manual*. Untuk operasi *manual*, *forward* dan *reverse* akan aktif jika tombol *forward* atau tombol *reverse manual* ditekan. Dan untuk menonaktifkannya bisa dengan menekan tombol stop. Untuk proses *auto*, *forward* dan *reverse* akan aktif sesuai dengan *setpoint* dan *durasi* yang telah ditentukan.



Gambar 17. Baris Program 14

o. Baris Program 15 – 19

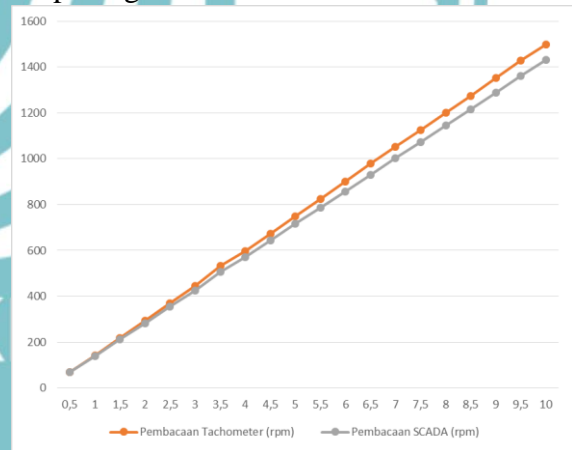
Baris program 15 berisi proses *feeding alarm abnormal* ke *alarm block*. Masing-masing *feed* akan aktif jika *setpoint mixing auto* tidak diisi.



Gambar 18. Baris Program 18

2. Pengujian Operasi Motor *Mixing* putaran *Forward* secara *Manual* Melalui *Potentiometer 1*

Hasil analisa operasi motor *mixing* putaran *forward* melalui *potentiometer 1* adalah berupa grafik pembacaan kecepatan putaran motor pada pengukuran melalui tachometer dan pembacaan pada perangkat SCADA.



Gambar 19. Grafik Pengujian Operasi Putaran *Forward* melalui *Potentiometer 1*

Hasil grafik pada gambar 19 dihasilkan nilai *error* pembacaan yang dihitung dengan persamaan (1).

$$Error (\%) = \frac{rpm\ SCADA - rpm\ tachometer}{rpm\ tachometer} \times 100\% \dots (1)$$

Pada grafik gambar 19 dihasilkan nilai terbesar di angka 5,01% pada pengoperasian *analog input* 6,5 V. penghitungan menggunakan persamaan 1, maka dihasilkan sebagai berikut :

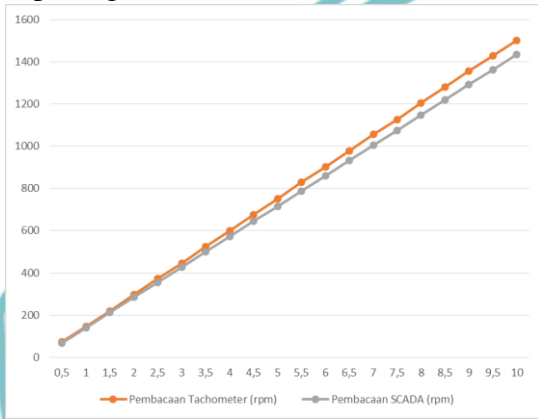
$$Error\ rpm (\%) = \frac{929 - 978}{929} \times 100\%$$

$$Error\ rpm (\%) = 5,01\%$$



3. Pengujian Operasi Motor *Mixing* putaran *Reverse* secara *Manual* Melalui *Potentiometer 1*

Hasil analisa operasi motor *mixing* putaran *reverse* melalui *potentiometer 1* adalah berupa grafik pembacaan kecepatan putaran motor pada pengukuran melalui tachometer dan pembacaan pada perangkat SCADA.



Gambar 20. Grafik Pengujian Operasi Putaran *Reverse* melalui *Potentiometer 1*

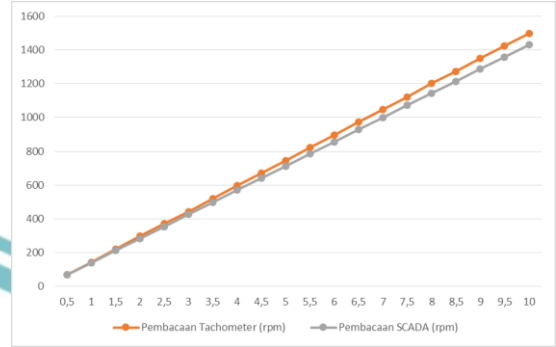
Hasil grafik pada gambar 20 dihasilkan nilai *error* pembacaan yang dihitung dengan persamaan (1). Pada grafik gambar dihasilkan nilai terbesar di angka 5,55% pada pengoperasian *analog input* 0,5 V. penghitungan menggunakan persamaan 1, maka dihasilkan sebagai berikut :

$$Error\ rpm\ (\%) = \frac{68 - 72}{68} \times 100\%$$

$$Error\ rpm\ (\%) = 5,55\ \%$$

4. Pengujian Operasi Motor *Mixing* putaran *Forward* secara *Manual* Melalui *Potentiometer 2*

Hasil analisa operasi motor *mixing* putaran *forward* melalui *potentiometer 2* adalah berupa grafik pembacaan kecepatan putaran motor pada pengukuran melalui tachometer dan pembacaan pada perangkat SCADA.



Gambar 21. Grafik Pengujian Operasi Putaran *Forward* melalui *Potentiometer 2*

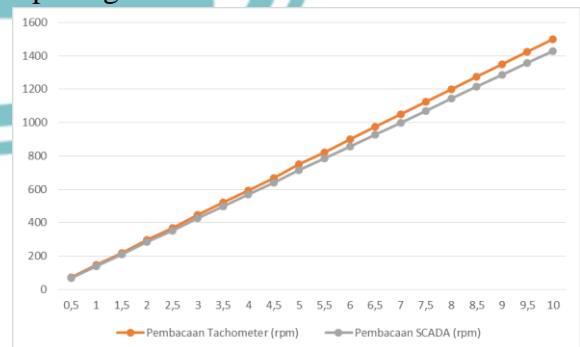
Hasil grafik pada gambar 21 dihasilkan nilai *error* pembacaan yang dihitung dengan persamaan (1). Pada grafik gambar dihasilkan nilai terbesar di angka 4,74% pada pengoperasian *analog input* 8 V. penghitungan menggunakan persamaan 1, maka dihasilkan sebagai berikut :

$$Error\ rpm\ (\%) = \frac{1144 - 1201}{1144} \times 100\%$$

$$Error\ rpm\ (\%) = 4,74\ \%$$

5. Pengujian Operasi Motor *Mixing* putaran *Reverse* secara *Manual* Melalui *Potentiometer 2*

Hasil analisa operasi motor *mixing* putaran *forward* melalui *potentiometer 2* adalah berupa grafik pembacaan kecepatan putaran motor pada pengukuran melalui tachometer dan pembacaan pada perangkat SCADA.



Gambar 22. Grafik Pengujian Operasi Putaran *Reverse* melalui *Potentiometer 2*



Hasil grafik pada gambar 22 dihasilkan nilai *error* pembacaan yang dihitung dengan persamaan (1). Pada grafik gambar dihasilkan nilai terbesar di angka 4,74% pada pengoperasian *analog input* 8 V. penghitungan menggunakan persamaan 1, maka dihasilkan sebagai berikut :

$$Error\ rpm\ (\%) = \frac{69 - 72,9}{72,9} \times 100\%$$

$$Error\ rpm\ (\%) = 5,34\ \%$$

KESIMPULAN

Dari penelitian dihasilkan bahwa nilai *error* bacaan kecepatan motor pada operasi *mixing* dihasilkan dengan nilai tertinggi pada angka 5,55%. Hasil pembacaan kecepatan putar motor dihasilkan bahwa kecepatan pada pengukuran lebih tinggi daripada nominal pada *nameplate* motor. Peningkatan akurasi

pembacaan kecepatan motor dapat dilakukan dengan menambahkan perangkat *rotary encoder* dengan sistem komunikasi serial.

DAFTAR PUSTAKA

- [1] M. F. E. N. H. A W NIENOW, *Mixing in the Process Industries: Second Edition*, Oxford: A division of Reed Educational and Professional Publishing Ltd, 2001.
- [2] M. M. Lashin, "DIFFERENT APPLICATIONS OF PROGRAMMABLE LOGIC CONTROLLER (PLC)," *International Journal of Computer Science, Engineering, and Information Technology (IJCEIT)*, vol. 4, no. 1, p. 27, 2014.

POLITEKNIK
NEGERI
JAKARTA