

BAB IV

PENGUJIAN DAN ANALISA

4.1. Implementasi Sistem

Pada tahap ini dilakukan implementasi seluruh rancangan sistem yang telah dijelaskan pada Bab III. Implementasi perangkat keras dan perangkat lunak yang saling terintegrasi sehingga mampu membentuk sistem pengendalian laju aliran air berbasis OpenPLC menggunakan metode PID. Implementasi perangkat keras mencakup proses perakitan seluruh komponen mulai dari panel kontrol, sistem perpipaan, sensor, aktuator, hingga catu daya. Sedangkan implementasi perangkat lunak meliputi pembuatan program OpenPLC, program mikrokontroler ESP32, Node-Red sebagai HMI, serta komunikasi antar perangkat.

4.1.1. Implementasi Perangkat Keras

Pada tahap implementasi perangkat keras dilakukan proses perakitan seluruh komponen yang digunakan pada penelitian sehingga membentuk suatu sistem pengendalian laju aliran air yang dapat beroperasi sesuai dengan rancangan. Implementasi dilakukan berdasarkan wiring diagram dan konektivitas antar perangkat yang telah dirancang pada Bab III. Sistem terdiri atas panel kontrol, tangki, pompa air, sensor *flow*, dua buah mikrokontroler ESP32, *driver* motor BTS7960, serta catu daya yang saling terhubung membentuk sistem kendali berbasis OpenPLC. Seluruh komponen tersebut dirangkai menjadi satu kesatuan sistem sehingga mampu menjalankan fungsi akuisisi data, pemrosesan sinyal, pengendalian aktuator, serta monitoring sistem secara *real-time*. Implementasi perangkat keras dilakukan secara bertahap dimulai dari pemasangan komponen mekanik, instalasi panel kontrol, pemasangan sensor dan aktuator, hingga proses pengkabelan sesuai dengan rancangan konektivitas yang telah dibuat. Setelah seluruh komponen terpasang, dilakukan pengujian awal untuk memastikan setiap perangkat dapat beroperasi sesuai dengan fungsi yang telah direncanakan sebelum diintegrasikan dengan perangkat lunak OpenPLC.

Selanjutnya setiap komponen dijelaskan berdasarkan fungsi dan implementasinya pada sistem.

- a. Foto keseluruhan alat

Gambar 4.1 menunjukkan hasil implementasi keseluruhan sistem pengendalian laju aliran air berbasis OpenPLC menggunakan metode PID. Sistem terdiri atas panel kontrol sebagai pusat kendali, dua buah tangki sebagai media sirkulasi air, pompa DC sebagai aktuator utama, sensor water flow sebagai pembaca laju aliran air, dua buah mikrokontroler ESP32, driver motor BTS7960, serta *Human Machine Interface* (HMI) berbasis Node-RED

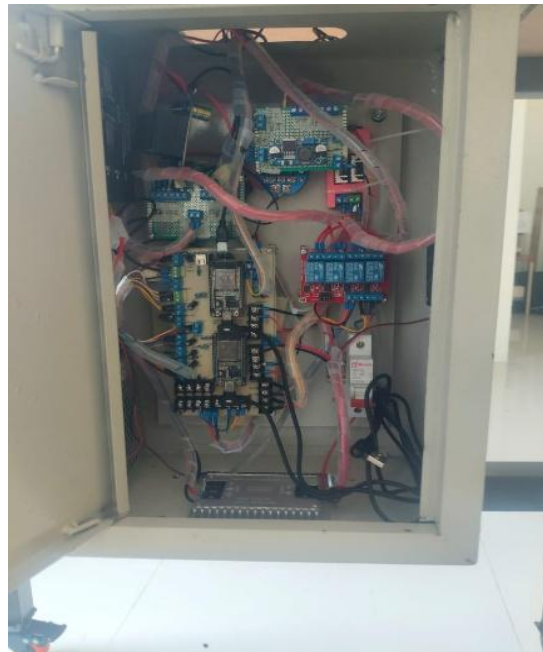


Gambar 4.1 Keseluruhan Alat

Implementasi keseluruhan alat dilakukan dengan mengintegrasikan seluruh komponen mekanik, elektronik, dan perangkat lunak sehingga membentuk satu sistem yang mampu melakukan proses pengendalian secara otomatis. Seluruh komponen dipasang pada rangka alat dengan mempertimbangkan kemudahan proses instalasi, pemeliharaan, serta keamanan selama proses pengujian. Posisi setiap komponen juga disesuaikan dengan alur aliran air dan jalur pengkabelan agar sistem bekerja secara optimal serta meminimalkan gangguan selama proses eksperimen.

b. Foto Panel Kontrol

Panel kontrol berfungsi sebagai pusat integrasi seluruh komponen kelistrikan pada sistem. Di dalam panel kontrol terdapat mikrokontroler ESP32, *driver* BTS7960, *power supply*, terminal *block*, serta rangkaian distribusi daya. Penggunaan panel kontrol bertujuan untuk meningkatkan kerapian instalasi, mempermudah proses pemeliharaan, serta memberikan perlindungan terhadap komponen elektronik dari gangguan lingkungan luar.



Gambar 4.2 Panel Kontrol

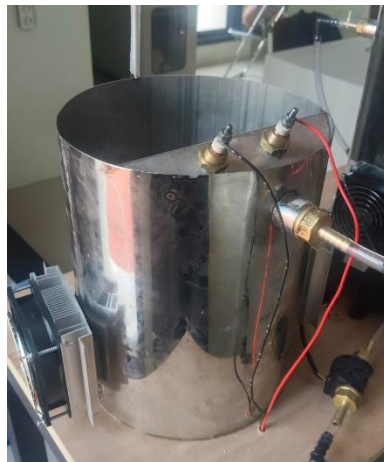
Seluruh komponen di dalam panel disusun berdasarkan fungsi masing-masing sehingga jalur distribusi daya dan jalur sinyal dapat dipisahkan dengan baik. Penyusunan tersebut bertujuan untuk mempermudah proses identifikasi apabila terjadi gangguan pada sistem serta memudahkan proses pengembangan apabila di kemudian hari dilakukan penambahan komponen. Selain itu, penggunaan terminal *block* juga mempermudah proses pemasangan maupun pelepasan komponen tanpa harus mengubah keseluruhan instalasi pengkabelan.

1. Tangki

Tangki digunakan sebagai media penyimpanan dan sirkulasi air selama proses pengujian berlangsung. Sistem menggunakan dua buah tangki, yaitu tangki penampung sebagai sumber air dan tangki proses sebagai tempat pengukuran laju aliran.



Gambar 4.3 Tangki 1 10 L



Gambar 4.4 Tangki 2 5 L

Sistem menggunakan dua buah tangki, yaitu tangki penampung sebagai sumber air dan tangki proses sebagai tempat pengukuran laju aliran. Air dipompa dari tangki penampung menuju tangki proses, kemudian dialirkan kembali sehingga membentuk sistem sirkulasi tertutup (*closed circulation system*).

2. Pompa DC

Pompa DC 12 Volt berfungsi sebagai aktuator utama yang mengatur besar kecilnya laju aliran air pada sistem.



Gambar 4.5 Pompa DC 12 V

Pompa DC 12 V digunakan sebagai aktuator utama pada sistem pengendalian laju aliran air. Besarnya kecepatan pompa diatur menggunakan Motor PWM yang dibentuk oleh ESP32 pendukung berdasarkan keluaran pengendali dari ESP32 OpenPLC. Keluaran CTRL_PID terlebih dahulu dikirim melalui jalur komunikasi PWM-RC-ADC menuju ESP32 pendukung. Nilai yang diterima kemudian digunakan sebagai Motor PWM pada driver BTS7960. Peningkatan Motor PWM meningkatkan tingkat aktiasi pompa sehingga laju aliran air yang dihasilkan juga meningkat.

3. Sensor *Flow*

Sensor *water flow* YF-S201 digunakan sebagai sensor utama untuk mengukur laju aliran air pada sistem.

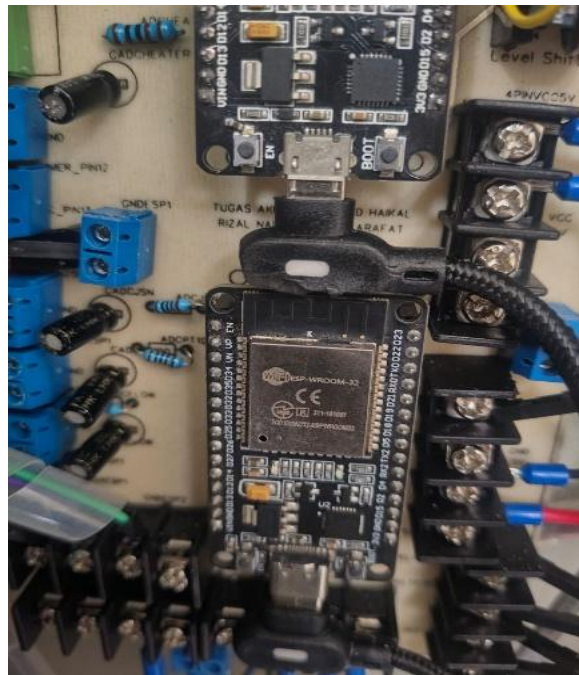


Gambar 4.6 Sensor Flow YF-S201

Sensor bekerja berdasarkan prinsip *Hall Effect* dengan menghasilkan pulsa digital yang jumlahnya sebanding dengan laju aliran air yang melewati sensor. Pulsa tersebut kemudian diproses oleh mikrokontroler ESP32 sehingga diperoleh nilai laju aliran dalam satuan mililiter per menit (mL/min). Nilai tersebut selanjutnya dikirimkan ke OpenPLC sebagai *Process Variable* (PV) pada proses pengendalian PID.

4. ESP32

Penelitian ini menggunakan dua buah mikrokontroler ESP32 yang memiliki fungsi berbeda. ESP32 pertama digunakan sebagai OpenPLC *Runtime* yang menjalankan logika PLC dan algoritma PID. ESP32 kedua bertugas melakukan pembacaan sensor serta mengendalikan perangkat pendukung.



Gambar 4.7 ESP32

Pembagian fungsi kedua ESP32 dilakukan untuk memisahkan proses eksekusi program pengendalian dengan proses akuisisi dan penerapan sinyal pada perangkat keras. ESP32 OpenPLC difokuskan untuk menjalankan OpenPLC Runtime, program *Function Block* Diagram, dan Function Block CTRL_PID. Sementara itu, ESP32 pendukung menangani pembacaan sensor YF-S201 serta penerapan keluaran kendali pada aktuator. Pertukaran data antara

kedua ESP32 dilakukan melalui jalur PWM–RC–ADC sesuai dengan rancangan komunikasi sistem.

5. *Driver* BTS7960

Driver BTS7960 digunakan sebagai antarmuka daya antara ESP32 pendukung dan pompa DC 12 V. *Driver* tidak menerima keluaran CTRL_PID secara langsung. Keluaran CTRL_PID dari ESP32 OpenPLC terlebih dahulu dikirim menuju ESP32 pendukung melalui jalur PWM–RC–ADC. Nilai yang diterima kemudian digunakan sebagai Motor PWM pada kanal RPWM BTS7960, sedangkan kanal LPWM dipertahankan pada kondisi rendah karena pompa hanya dioperasikan dalam satu arah.

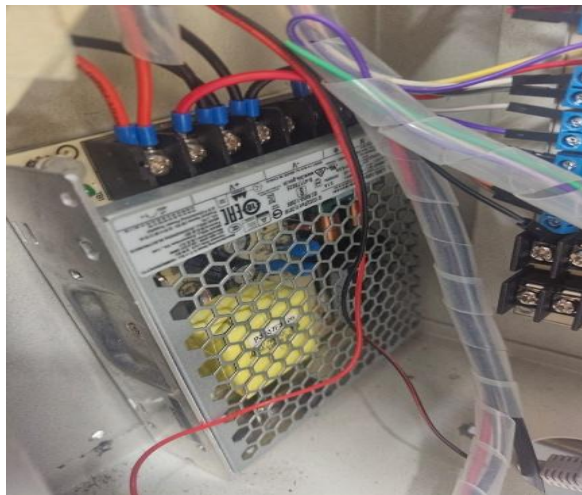


Gambar 4.8 Driver BTS7960

Pemilihan BTS7960 didasarkan pada kemampuan driver dalam menangani arus yang relatif besar sehingga sesuai digunakan untuk mengendalikan pompa DC yang digunakan pada penelitian ini. Selain itu, penggunaan driver BTS7960 memungkinkan pengaturan kecepatan pompa dilakukan secara halus (*smooth*) melalui variasi *duty cycle* PWM yang dihasilkan oleh pengendali PID.

6. *Power Supply*

Power supply berfungsi sebagai sumber catu daya bagi seluruh komponen pada sistem. Tegangan keluaran didistribusikan sesuai kebutuhan masing-masing perangkat, seperti mikrokontroler ESP32, sensor, driver BTS7960, dan pompa DC.

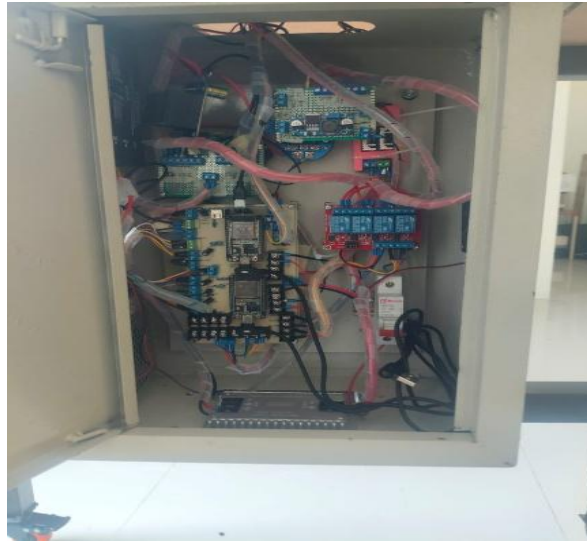


Gambar 4.9 Power Supply

Penggunaan catu daya yang stabil sangat penting untuk menjaga performa sistem selama proses pengujian berlangsung.

c. Wiring Sistem

Implementasi *wiring* dilakukan berdasarkan rancangan konektivitas yang telah dibuat pada Bab III. Seluruh sensor, aktuator, mikrokontroler, driver motor, dan catu daya dihubungkan menggunakan terminal *block* sehingga menghasilkan instalasi yang rapi, mudah dilakukan pemeliharaan, serta meminimalkan kesalahan pemasangan selama proses implementasi.



Gambar 4.10 Wiring Sistem

Proses pengkabelan dilakukan dengan memperhatikan aspek keamanan, kerapian, serta kemudahan pemeliharaan. Jalur sinyal dan jalur catu daya dipisahkan untuk meminimalkan gangguan elektromagnetik yang dapat mempengaruhi pembacaan sensor maupun proses pengendalian. Setelah seluruh *wiring* selesai dipasang, dilakukan pengecekan kontinuitas dan pengujian fungsi pada setiap jalur koneksi untuk memastikan tidak terjadi kesalahan pemasangan sebelum sistem dioperasikan secara keseluruhan.

4.1.2. Hasil Implementasi Perangkat Lunak

Hasil implementasi perangkat lunak menunjukkan bahwa rancangan program pada Bab III dapat dijalankan pada sistem. Rancangan Function Block Diagram OpenPLC, program ESP32 pada Arduino IDE, pengolahan sensor, komunikasi pin-to-pin, dan *Function Block* CTRL_PID telah dipindahkan ke Bab III karena bagian tersebut merupakan tahapan perancangan. Bab IV selanjutnya difokuskan pada hasil implementasi, verifikasi, pengujian, dan analisa kinerja sistem.

4.2. Pengujian Perangkat

4.2.1. Hasil Pengujian Akurasi Sistem Pembacaan Sensor *Water Flow* YF-S201

Pengujian dilakukan untuk mengetahui kesesuaian nilai pembacaan sistem *water flow* terhadap volume aktual yang diukur menggunakan gelas ukur. Gelas

ukur digunakan sebagai acuan volumetrik, sedangkan nilai pada sistem digunakan sebagai hasil pembacaan sensor. Pengujian dilakukan pada lima titik volume untuk melihat besar selisih dan persentase error pada beberapa kondisi operasi.

Tabel 4.1 Hasil Pengujian Respons Sensor Flow

Pembacaan Sistem	Gelas Ukur	Selisih Absolut	error
1000	1050	50	4,76
2000	1950	50	2,56
3000	2850	150	5,26
4000	3850	150	3,90
5000	4900	100	2,04

Berdasarkan Tabel 4.1, terdapat perbedaan antara pembacaan sistem water flow dan hasil pengukuran volumetrik menggunakan gelas ukur. Pada pembacaan 1000 mL, volume aktual yang diperoleh sebesar 1050 mL dengan error 4,76%. Pada pembacaan 2000 mL diperoleh volume aktual sebesar 1950 mL dengan error 2,56%. Selanjutnya, pada pembacaan 3000 mL dan 4000 mL diperoleh volume aktual masing-masing sebesar 2850 mL dan 3850 mL dengan error 5,26% dan 3,90%. Pada pembacaan 5000 mL diperoleh volume aktual sebesar 4900 mL dengan error 2,04%.

Dari lima titik pengujian diperoleh rata-rata error sebesar 3,70% dengan rata-rata selisih absolut sebesar 100 mL. Error terbesar terjadi pada titik 3000 mL, yaitu sebesar 5,26%, sedangkan error terkecil diperoleh pada titik 5000 mL sebesar 2,04%. Hasil tersebut menunjukkan bahwa pembacaan sistem memiliki kecenderungan yang mengikuti perubahan volume aktual, meskipun masih terdapat penyimpangan pada masing-masing titik pengujian.

Perbedaan nilai pembacaan dapat dipengaruhi oleh karakteristik sensor YF-S201 yang menghasilkan data berbasis pulsa, resolusi jumlah pulsa selama interval pengukuran, serta kondisi aliran air pada saat pengujian. Oleh karena itu, hasil pengujian volumetrik ini digunakan untuk mengetahui kesesuaian pembacaan

sistem terhadap kondisi aktual, sedangkan evaluasi performa pengendali PID pada pengujian closed-loop tetap menggunakan *process variable* yang diproses oleh OpenPLC.

4.2.2. Pengujian Driver BTS7960

Pengujian BTS7960 dilakukan dengan memberikan beberapa nilai Motor PWM pada kanal RPWM, sedangkan LPWM dipertahankan nol. Hasil pengamatan menunjukkan bahwa peningkatan Motor PWM meningkatkan kecepatan pompa dan laju aliran yang dihasilkan. Hal ini mengonfirmasi bahwa BTS7960 dapat meneruskan perintah aktuasi dari ESP32 pendukung ke pompa.

4.2.3. Pengujian Sensor JSN-SR04T

1. Tujuan Pengujian

Pengujian sensor ultrasonik JSN-SR04T dilakukan untuk mengetahui kemampuan sensor dalam mengukur jarak antara posisi sensor dengan permukaan air pada tangki. Sensor dipasang pada bagian atas tangki dengan posisi tetap terhadap dasar tangki, sehingga perubahan ketinggian permukaan air akan menyebabkan perubahan nilai jarak yang terbaca oleh sensor.

Pada kondisi tangki kosong, jarak antara sensor dengan dasar tangki berada pada kisaran 39-40cm. Ketika air mengisi tangki, permukaan air akan semakin mendekati sensor sehingga nilai jarak yang terbaca menjadi semakin kecil. Oleh karena itu, nilai pembacaan sensor dapat digunakan sebagai informasi untuk mengetahui kondisi permukaan air di dalam tangki.

Pada penelitian ini, nilai yang ditampilkan pada HMI merupakan jarak antara sensor dan permukaan air, bukan tinggi air secara langsung. Ketinggian air dapat diperoleh dari selisih antara posisi sensor terhadap dasar tangki dan jarak yang dibaca sensor. Informasi tersebut digunakan sebagai variabel pendukung dalam sistem, sedangkan variabel utama yang dikendalikan menggunakan PID tetap berupa laju aliran air.

2. Metode Pengujian

Pengujian dilakukan dengan memberikan beberapa variasi volume air pada tangki, yaitu 1000mL, 2000mL, 3000mL, 4000mL, 5000mL, dan

6000mL. pada setiap kondisi volume, ketinggian permukaan air dari dasar tangki diukur secara langsung menggunakan alat ukur sebagai nilai referensi.

Posisi sensor JSN terhadap dasar tangki pada pengujian adalah sebesar 40 cm. Karena sensor dipasang pada bagian atas tangki, sensor membaca jarak antara posisi sensor dengan permukaan air. Oleh karena itu, jarak referensi antara sensor dan permukaan air diperoleh dari selisih antara tinggi posisi sensor terhadap dasar tangki dengan tinggi permukaan air.

Pada volume 1.000 mL, tinggi air yang diperoleh sebesar 2,5 cm, sehingga jarak referensinya adalah:

$$D_{ref} = 40 - 2,5$$

$$D_{ref} = 37,5 \text{ cm}$$

Pada volume 2.000 mL, tinggi air yang diperoleh sebesar 5,5 cm, sehingga:

$$D_{ref} = 40 - 5,5$$

$$D_{ref} = 34,5 \text{ cm}$$

Pada volume 3.000 mL, tinggi air yang diperoleh sebesar 8,2 cm, sehingga:

$$D_{ref} = 40 - 8,2$$

$$D_{ref} = 31,8 \text{ cm}$$

Pada volume 4.000 mL, tinggi air yang diperoleh sebesar 10,8 cm, sehingga:

$$D_{ref} = 40 - 10,8$$

$$D_{ref} = 29,2 \text{ cm}$$

Pada volume 5.000 mL, tinggi air yang diperoleh sebesar 13,5 cm, sehingga:

$$D_{ref} = 40 - 13,5$$

$$D_{ref} = 26,5 \text{ cm}$$

Pada volume 6.000 mL, tinggi air yang diperoleh sebesar 16 cm, sehingga:

$$D_{ref} = 40 - 16$$

$$D_{ref} = 24 \text{ cm}$$

Tabel 4.2 Hasil Pengujian Sensor JSN-SR04T

No	Volume (ml)	Tinggi Air (cm)	Jarak Referensi JSN (cm)	Pembacaan JSN (cm)	Selisih (cm)	Error (%)
1	1000	2,5	37,5	37	0,5	1,33
2	2000	5,5	34,5	34	0,5	1,45
3	3000	8,2	31,8	33	1,2	3,77
4	4000	10,8	29,2	30	0,8	2,74
5	5000	13,5	26,5	27	0,5	1,89
6	6000	16	24,0	25	1,0	4,17

Pengujian sensor JSN-SR04T dilakukan dengan memberikan beberapa variasi volume air pada tangki dan mengukur tinggi permukaan air menggunakan penggaris sebagai referensi. Karena sensor dipasang di bagian atas tangki, pembacaan sensor berupa jarak dari sensor ke permukaan air. Oleh karena itu, nilai referensi dikonversi ke jarak sensor–permukaan sebelum dibandingkan dengan hasil pembacaan JSN-SR04T.

$$D_{ref} = 40 - 2,5$$

$$D_{ref} = 37,5 \text{ cm}$$

Pada volume air 1000 mL, diperoleh ketinggian air sebesar 2,5 cm, sehingga jarak referensi antara sensor dan permukaan air adalah 37,5 cm. hasil pembacaan sensor jsn pada kondisi tersebut sebesar 37 cm, sehingga selisih pembacaan adalah 0,5 cm, dan persentase error diperoleh sebesar 1, 33%. Perhitungan yang sama dilakukan pada seluruh variasi volume air.

Berdasarkan hasil pengujian, nilai error terkecil diperoleh sebesar 1,33%, sedangkan nilai error terbesar sebesar 4,17%. rata rata error pembacaan sensor pada enam kondisi pengujian adalah sebesar 2, 56%.

Hasil pengujian menunjukkan bahwa perubahan volume air menghasilkan perubahan jarak pembacaan sensor yang sesuai dengan kondisi fisik tangki. Ketika volume air bertambah, ketinggian permukaan air meningkat sehingga jarak

antara sensor dan permukaan air menjadi semakin kecil. Hal ini terlihat dari nilai pembacaan sensor yang menurun seiring dengan bertambahnya volume air.

Meskipun terdapat selisih antara nilai pembacaan sensor dan jarak referensi, nilai error yang diperoleh masih relatif kecil pada seluruh kondisi pengujian. Dengan demikian, sensor JSN-SR04T dapat digunakan untuk memberikan informasi mengenai perubahan posisi permukaan air pada tangki. Pada sistem yang digunakan, nilai yang ditampilkan pada HMI merupakan jarak antara sensor dan permukaan air

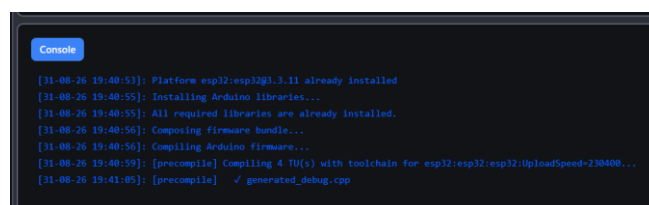
4.2.4. Pengujian ESP32 dan OpenPLC

Pengujian ESP32 dan OpenPLC dilakukan untuk memastikan bahwa mikrokontroler ESP32 dapat menjalankan program pengendalian yang dikembangkan menggunakan OpenPLC serta mampu melakukan pembacaan dan pembaruan variabel sesuai dengan kondisi sistem. Pengujian ini dilakukan sebelum pengujian *open loop* dan *closed-loop* untuk memastikan bahwa fungsi dasar perangkat pengendali telah berjalan dengan baik.

Pengujian meliputi proses kompilasi dan pengunggahan program, pengoperasian OpenPLC *Runtime*, pengamatan perubahan nilai variabel, serta verifikasi komunikasi antara program OpenPLC dengan input dan output yang digunakan pada sistem.

a. Pengujian Proses Kompilasi dan Upload Program

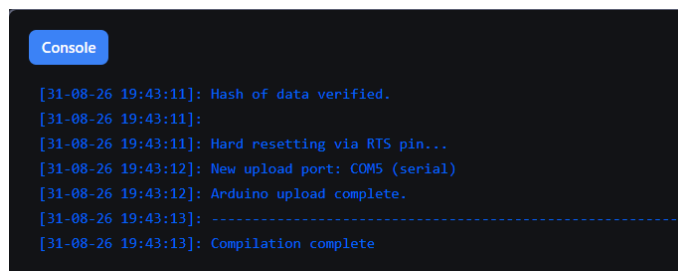
Pengujian pertama dilakukan dengan mengunggah program hasil perancangan OpenPLC menuju ESP32. Program yang digunakan berisi pemetaan variabel input dan output serta logika kontrol dasar yang digunakan pada sistem.



```

[31-08-26 19:40:53]: Platform esp32:esp32@1.11 already installed
[31-08-26 19:40:55]: Installing Arduino libraries...
[31-08-26 19:40:55]: All required libraries are already installed.
[31-08-26 19:40:55]: Composing firmware bundle...
[31-08-26 19:40:56]: Compiling Arduino firmware...
[31-08-26 19:40:59]: [precompile] Compiling 4 files with toolchain for esp32:esp32:esp32@1.11...
[31-08-26 19:41:05]: [precompile] ✓ generated_debug.cpp
  
```

Gambar 4.11 Proses Kompilasi Program OpenPLC untuk ESP32



```

[31-08-26 19:43:11]: Hash of data verified.
[31-08-26 19:43:11]:
[31-08-26 19:43:11]: Hard resetting via RTS pin...
[31-08-26 19:43:12]: New upload port: COM5 (serial)
[31-08-26 19:43:12]: Arduino upload complete.
[31-08-26 19:43:13]: -----
[31-08-26 19:43:13]: Compilation complete

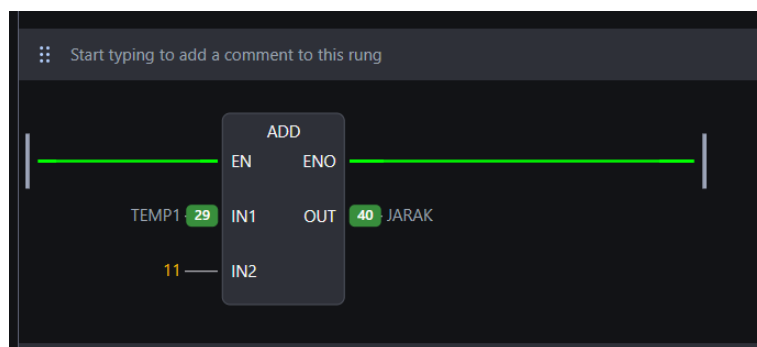
```

Gambar 4.12 Hasil Upload Program OpenPLC ke ESP32

Setelah proses kompilasi dan pengunggahan selesai, program dapat dijalankan pada ESP32 tanpa terjadi kesalahan pada proses upload. Hal ini menunjukkan bahwa program yang dibuat pada OpenPLC dapat diterapkan pada perangkat ESP32 yang digunakan sebagai pengendali.

b. Rancangan Pengujian OpenPLC *Runtime*

Pengujian selanjutnya dilakukan untuk memastikan bahwa OpenPLC Runtime dapat menjalankan program kontrol pada ESP32. Setelah program dijalankan, nilai variabel pada sistem diamati untuk memastikan bahwa program dieksekusi secara berulang.



Gambar 4.13 Kondisi OpenPLC saat Program Berjalan

Hasil pengujian menunjukkan bahwa OpenPLC Runtime dapat menjalankan program pada ESP32 dan memperbarui nilai variabel sesuai dengan kondisi input yang diberikan. Dengan demikian, ESP32 dapat digunakan sebagai perangkat keras untuk menjalankan fungsi pengendalian menggunakan OpenPLC.

c. Pengujian Perubahan Variabel

Pengujian perubahan variabel dilakukan untuk memastikan bahwa nilai yang digunakan dalam program OpenPLC dapat berubah sesuai kondisi sistem. Pengujian dilakukan dengan memberikan perubahan pada nilai masukan kemudian mengamati perubahan variabel yang terkait.



Gambar 4.14 Perubahan Variabel pada OpenPLC

Sebagai contoh, perubahan nilai laju aliran yang dibaca sensor menyebabkan nilai variabel *flow rate* pada sistem ikut berubah. Demikian pula perubahan nilai *setpoint* pada HMI dapat diterima oleh program kontrol. Hasil pengujian menunjukkan bahwa variabel input dan output dapat diperbarui sesuai dengan kondisi sistem sehingga dapat digunakan dalam proses pengendalian.

4.2.5. Verifikasi Implementasi OpenPLC Berbasis ESP32

Setelah fungsi dasar ESP32 dan OpenPLC berhasil diuji, selanjutnya dilakukan verifikasi implementasi OpenPLC pada keseluruhan sistem pengendalian laju aliran air. Verifikasi dilakukan untuk memastikan bahwa pemetaan variabel, eksekusi program kontrol, pembacaan variabel proses, serta keluaran pengendalian bekerja sesuai dengan rancangan.

Pada penelitian ini, OpenPLC Runtime dijalankan pada ESP32 utama dan digunakan pusat pengendalian sistem. Nilai laju aliran digunakan sebagai process variable, nilai laju aliran yang diinginkan digunakan sebagai *setpoint*, sedangkan keluaran kontrol digunakan sebagai dasar pengaturan PWM pompa.

a. Verifikasi Pemetaan Input dan Output

Verifikasi pemetaan dilakukan untuk memastikan bahwa variabel pada program OpenPLC telah terhubung dengan input dan output yang sesuai dengan rancangan sistem.

Tabel 4.3 Verifikasi Pemetaan Variabel OpenPLC

Variabel	Fungsi	Jenis	Alamat	Hasil
<i>Flow</i>	Laju Aliran Aktual	Input/PV	%MD202	Sesuai
Level	Level aktual	Input/ <i>Interlock</i>	%MD206	Sesuai
Setpoint	Nilai yang diinginkan	Input	%MW4 %MW6	Sesuai
PID OUT	Keluaran CTRL PID	Output	-	Sesuai
PWM OUT	Kendali pompa	Output	%QW0	sesuai

Variabel input digunakan untuk menerima data kondisi sistem, sedangkan variabel output digunakan untuk menghasilkan sinyal kendali terhadap aktuator. Hasil verifikasi menunjukkan bahwa perubahan pada input dapat dibaca oleh program dan perubahan output dapat diteruskan menuju bagian sistem yang sesuai.

b. Verifikasi Eksekusi *Function Block*

Verifikasi berikutnya dilakukan terhadap Function Block yang digunakan pada program OpenPLC. Pengujian dilakukan dengan mengubah nilai masukan kemudian mengamati perubahan kelaoran blok pada saat program dijalankan.

Pada kondisi operasi, program OpenPLC dieksekusi secara siklik. Nilai variabel proses diperbarui kemudian digunakan oleh blok fungsi untuk menghasilkan nilai keluaran berdasarkan logika program yang telah dibuat.

Untuk blok CTRL_PID , nilai setpoint dan nilai aktual digunakan sebagai masukan kontrol, sedangkan parameter KP, TN, dan TV digunakan untuk menentukan karakteristik pengendalian. Keluaran blok kemudian digunakan sebagai dasar nilai kendali pompa.

c. Verifikasi Variabel Proses, Setpoint, dan Output PID

Pengujian ini dilakukan dengan mengamati tiga variabel utama, yaitu:

SP, PV, Y

Dengan:

SP : Setpoint

PV : Laju Aliran Aktual

Y : Keluaran CTRL_PID

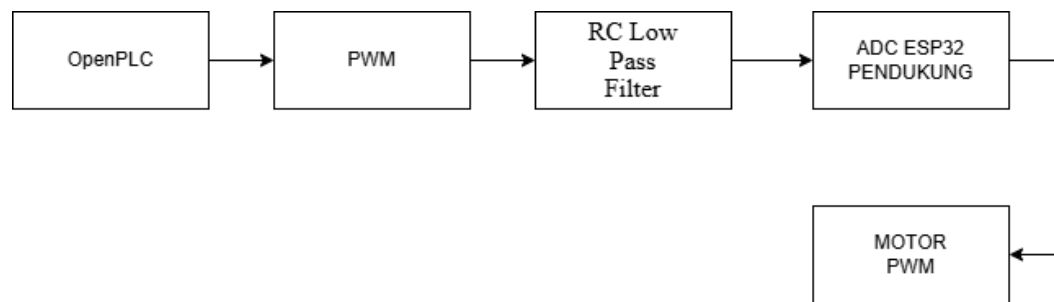
Pada saat nilai aktual berbeda dengan setpoint, OpenPLC menghasilkan keluaran kontrol berdasarkan nilai error. Perubahan keluaran kontrol selanjutnya digunakan dalam pengaturan PWM Pompa. Dengan demikian, perubahan nilai variabel proses dapat menghasilkan perubahan aksi pengendalian sesuai dengan program yang dirancang.

4.3. Pengujian Komunikasi *Pin-to-Pin* Antar ESP32

Pengujian komunikasi pin-to-pin untuk mengetahui kemampuan jalur komunikasi antara ESP32 yang menjalankan OpenPLC dengan ESP32 pendukung dalam meneruskan nilai keluaran. Pada sistem yang dirancang, nilai keluaran dari OpenPLC tidak dikirim dalam bentuk komunikasi serial atau jaringan, melainkan direpresentasikan dalam bentuk sinyal PWM melalui GPIO.

Sinyal PWM dari ESP32 OpenPLC selanjutnya dilewatkan melalui rangkaian RC *low-pass* filter sehingga diperoleh tegangan analog yang dapat dibaca oleh ADC pada ESP32 pendukung. Nilai ADC yang diterima kemudian digunakan sebagai nilai Motor PWM untuk mengatur kecepatan pompa melalui driver BTS7960.

Dengan demikian, jalur komunikasi yang diuji dapat digambarkan sebagai:



Gambar 4.15 Diagram Blok Jalur Komunikasi Pin To Pin

Pengujian dilakukan menggunakan empat kondisi keluaran, yaitu PWM 0, 22.000, 32.000, dan 44.000. Pada setiap kondisi dilakukan pengukuran tegangan setelah rangkaian RC low-pass filter menggunakan multimeter dan pengamatan nilai PID Flow ADC serta Motor PWM melalui Serial Monitor ESP32 pendukung.

Tujuan pengujian ini bukan untuk memperoleh nilai numerik yang sama antara keluaran OpenPLC dengan ADC penerima, karena kedua nilai menggunakan rentang yang berbeda. Pengujian lebih difokuskan untuk mengetahui apakah kenaikan keluaran PWM pada OpenPLC menghasilkan kenaikan tegangan pada jalur komunikasi dan diikuti dengan peningkatan nilai ADC pada ESP32 pendukung.

4.3.1. Pengujian Pengiriman PWM 0

Pengujian pertama dilakukan dengan memberikan nilai keluaran PWM sebesar **0** pada ESP32 OpenPLC. Kondisi ini digunakan sebagai titik awal untuk mengetahui kondisi jalur komunikasi ketika tidak terdapat perintah kendali terhadap pompa.

Sinyal keluaran diukur setelah melewati rangkaian RC low-pass filter menggunakan multimeter dalam mode tegangan DC. Hasil pengukuran ditunjukkan pada Gambar 4.15.



Gambar 4.16 Hasil Pengukuran Tegangan Keluaran RC Low-Pass Filter pada PWM 0

Berdasarkan Gambar 4.15, meskipun nilai PWM yang diberikan sebesar 0, multimeter masih menunjukkan tegangan sekitar 0,0924 V. Nilai tersebut sangat kecil dan berada mendekati kondisi 0 V.

Munculnya tegangan kecil pada kondisi PWM 0 dapat disebabkan oleh tegangan sisa pada rangkaian, noise elektronik, karakteristik kapasitor pada RC filter, sambungan kabel, maupun keterbatasan alat ukur. Oleh karena itu, keluaran analog tidak selalu menunjukkan tepat 0,000 V meskipun sinyal kendali berada pada kondisi minimum.

Setelah sinyal tersebut masuk ke ADC ESP32 pendukung, nilai yang diterima diamati melalui Serial Monitor seperti ditunjukkan pada Gambar 4.16.

```

pidADC = 0 | motorPWM = 0

=====
Distance       : 40.06 cm
PWM Distance   : 819
=====
ADC PT100      : 1085
Temperature    : 17.17 C
=====
Flow Pulse     : 0
Flow Rate      : 0.00 L/min
PWM Flow       : 0
=====
PID Flow ADC   : 0
Motor PWM      : 0
=====
PID Heater ADC : 17
Heater Power   : 0 W

```

Gambar 4.17 Hasil Pembacaan Serial Monitor pada Pengiriman PWM 0

Berdasarkan hasil pembacaan Serial Monitor, diperoleh:

PID Flow ADC=0

Dan

Motor PWM=0

Hasil tersebut menunjukkan bahwa tegangan sisa sebesar 0,0924 V tidak menyebabkan sistem menghasilkan perintah penggerakan pompa. Pada sisi perangkat lunak, kondisi tersebut tetap diterjemahkan sebagai nilai kendali nol.

Dengan demikian, jalur komunikasi dapat membedakan kondisi ketika tidak terdapat perintah keluaran. Hal ini penting karena pada kondisi keluaran PID bernilai nol, pompa juga harus menerima nilai kendali nol agar tidak beroperasi tanpa adanya perintah dari OpenPLC.

4.3.2. Pengujian Pengiriman PWM 22.000

Pengujian kedua dilakukan dengan menaikkan nilai keluaran OpenPLC menjadi 22.000. Pengujian ini bertujuan untuk mengetahui apakah kenaikan nilai PWM dari OpenPLC dapat menghasilkan kenaikan tegangan setelah RC low-pass filter dan selanjutnya dapat dibaca oleh ADC ESP32 pendukung.

Hasil pengukuran tegangan ditunjukkan pada Gambar 4.17.



Gambar 4.18 Hasil Pengukuran Tegangan Keluaran RC Low-Pass Filter pada PWM 22.000

Berdasarkan Gambar 4.17, tegangan yang diperoleh setelah RC low-pass filter sebesar 1,113 V. Jika dibandingkan dengan kondisi PWM 0 yang hanya

menghasilkan tegangan sekitar 0,0924 V, terjadi peningkatan tegangan yang jelas ketika nilai PWM dinaikkan menjadi 22.000.

Hasil tersebut menunjukkan bahwa perubahan *duty cycle* PWM dari ESP32 OpenPLC berhasil diubah oleh RC *low-pass* filter menjadi perubahan tegangan rata-rata. Tegangan tersebut selanjutnya diterima oleh ADC ESP32 pendukung. Hasil pembacaan pada Serial Monitor ditunjukkan pada Gambar 4.18.

```
pidADC = 1369 | motorPWM = 1369

=====
Distance      : 39.08 cm
PWM Distance  : 798
=====
ADC PT100     : 1157
Temperature   : 20.80 C
=====
Flow Pulse    : 10
Flow Rate     : 1.33 L/min
PWM Flow      : 1556
=====
PID Flow ADC  : 1369
Motor PWM     : 1369
=====
PID Heater ADC : 0
Heater Power  : 0 %
```

Gambar 4.19. Hasil Pembacaan Serial Monitor pada Pengiriman PWM 22.000

Berdasarkan Gambar 4.18, diperoleh nilai:

PID Flow ADC=1369

Dan

Motor PWM=1369

Nilai tersebut menunjukkan bahwa perubahan tegangan pada jalur pin-to-pin berhasil diterima oleh ESP32 pendukung. Nilai PID Flow ADC kemudian diteruskan sebagai Motor PWM, sehingga keluaran kendali yang berasal dari OpenPLC dapat digunakan untuk mengatur pompa.

Pada kondisi ini juga terlihat bahwa nilai 22.000 dari OpenPLC tidak berubah menjadi nilai 22.000 pada ESP32 pendukung. Hal tersebut memang sesuai dengan rancangan komunikasi karena data tidak dipindahkan dalam bentuk angka digital secara langsung.

Nilai dari OpenPLC terlebih dahulu menentukan *duty cycle* PWM, kemudian sinyal PWM diratakan menjadi tegangan analog oleh RC low-pass filter. Tegangan tersebut selanjutnya dikonversi kembali menjadi nilai ADC pada ESP32 pendukung. Oleh karena itu, nilai pada sisi pengirim dan penerima mempunyai skala yang berbeda.

Hal yang menjadi indikator utama keberhasilan komunikasi adalah arah perubahan nilai yang konsisten, yaitu ketika nilai keluaran OpenPLC meningkat, tegangan dan ADC penerima juga meningkat.

4.3.3. Pengujian Pengiriman PWM 32.000

Pengujian ketiga dilakukan menggunakan keluaran PWM sebesar 32.000. Nilai tersebut lebih tinggi dibandingkan kondisi sebelumnya sehingga tegangan setelah RC filter dan nilai ADC penerima diharapkan mengalami peningkatan. Hasil pengukuran tegangan ditunjukkan pada Gambar 4.19.



Gambar 4.20 Hasil Pengukuran Tegangan Keluaran RC Low-Pass Filter pada PWM 32.000

Berdasarkan hasil pengukuran, tegangan setelah RC *low-pass* filter sebesar 1,594 V. Nilai tersebut lebih tinggi daripada tegangan 1,113 V yang diperoleh ketika keluaran PWM sebesar 22.000.

Hasil ini memperlihatkan bahwa perubahan PWM dari 22.000 menjadi 32.000 menghasilkan kenaikan tegangan pada jalur komunikasi. Dengan demikian, rangkaian RC filter tidak hanya mampu membedakan kondisi minimum dan

maksimum, tetapi juga mampu menghasilkan tingkat tegangan yang berbeda untuk nilai PWM antara. Hasil pembacaan pada ESP32 pendukung ditunjukkan pada Gambar 4.20.

```

pidADC = 1877 | motorPWM = 1877
=====
Distance      : 37.47 cm
PWM Distance   : 757
=====
ADC PT100     : 1149
Temperature    : 20.39 C
=====
Flow Pulse    : 18
Flow Rate     : 2.40 L/min
PWM Flow      : 2808
=====
PID Flow ADC   : 1877
Motor PWM      : 1877
=====
PID Heater ADC : 0
Heater Power   : 0 %

```

Gambar 4.21. Hasil Pembacaan Serial Monitor pada Pengiriman PWM 32.000

Berdasarkan Serial Monitor, diperoleh:

PID Flow ADC=1877

Dan

Motor PWM=1877

Jika dibandingkan dengan kondisi PWM 22.000, nilai PID Flow ADC meningkat dari 1369 menjadi 1877. Peningkatan tersebut searah dengan perubahan tegangan hasil RC filter dari 1,113 V menjadi 1,594 V.

Dengan demikian, terdapat hubungan yang konsisten antara perubahan keluaran OpenPLC, tegangan pada jalur komunikasi, dan nilai yang diterima ESP32 pendukung.

Pada kondisi aktual, kenaikan ADC tidak harus mempunyai hubungan linear sempurna dengan perubahan tegangan. Hal tersebut disebabkan karakteristik ADC ESP32 yang tidak ideal, toleransi rangkaian RC, variasi tegangan referensi, noise, dan perubahan kecil pada kondisi sambungan selama proses pengukuran.

Meskipun demikian, hasil yang diperoleh masih menunjukkan pola perubahan yang jelas dan searah. Oleh karena itu, komunikasi masih dapat dianggap bekerja sesuai fungsi yang dirancang.

4.3.4. Pengujian Pengiriman PWM 44.000

Pengujian selanjutnya dilakukan menggunakan nilai keluaran PWM sebesar 44.000. Kondisi ini digunakan untuk melihat kemampuan komunikasi ketika nilai kendali ditingkatkan menuju daerah keluaran tinggi.

Hasil pengukuran tegangan setelah RC *low-pass* filter ditunjukkan pada Gambar 4.21.



Gambar 4.22 Hasil Pengukuran Tegangan Keluaran RC Low-Pass Filter pada PWM 44.000

Berdasarkan hasil pengukuran, tegangan yang diperoleh sebesar 2,150 V. Nilai tersebut menjadi tegangan tertinggi dari empat kondisi pengujian yang dilakukan.

Jika dibandingkan dengan pengujian sebelumnya, terjadi peningkatan tegangan secara bertahap dari 0,0924 V pada PWM 0, kemudian 1,113 V pada PWM 22.000, 1,594 V pada PWM 32.000, dan menjadi 2,150 V pada PWM 44.000.

Pola tersebut menunjukkan bahwa tegangan rata-rata hasil RC filter mengikuti perubahan keluaran PWM yang diberikan oleh ESP32 OpenPLC.

Hasil pembacaan pada sisi ESP32 pendukung ditunjukkan pada Gambar 4.22.

```

pidADC = 2661 | motorPWM = 2661
=====
Distance      : 37.47 cm
PWM Distance   : 757
=====
ADC PT100     : 1147
Temperature    : 20.29 C
=====
Flow Pulse    : 22
Flow Rate     : 2.93 L/min
PWM Flow      : 3428
=====
PID Flow ADC   : 2661
Motor PWM      : 2661
=====
PID Heater ADC : 0
Heater Power   : 0 %

```

Gambar 4.23 Hasil Pembacaan Serial Monitor pada Pengiriman PWM 44.000

Berdasarkan serial monitor diperoleh:

PID FLOW ADC = 2661

Dan

Motor PWM = 2661

Nilai tersebut merupakan pembacaan terbesar dari seluruh kondisi yang digunakan dalam pengujian komunikasi. Jika dibandingkan dengan kondisi PWM 32.000, nilai ADC meningkat dari 1877 menjadi 2661, sedangkan tegangan meningkat dari 1,594 V menjadi 2,150 V. Hasil tersebut semakin memperkuat bahwa perubahan nilai pada OpenPLC dapat diteruskan melalui jalur komunikasi pin-to-pin dan diterima sebagai perubahan nilai yang searah oleh ESP32 pendukung.

Nilai Motor PWM yang sama dengan PID Flow ADC juga menunjukkan bahwa pada tahapan ini nilai yang diterima oleh ADC diteruskan menjadi nilai kendali aktuator sesuai dengan logika yang diterapkan pada ESP32 pendukung.

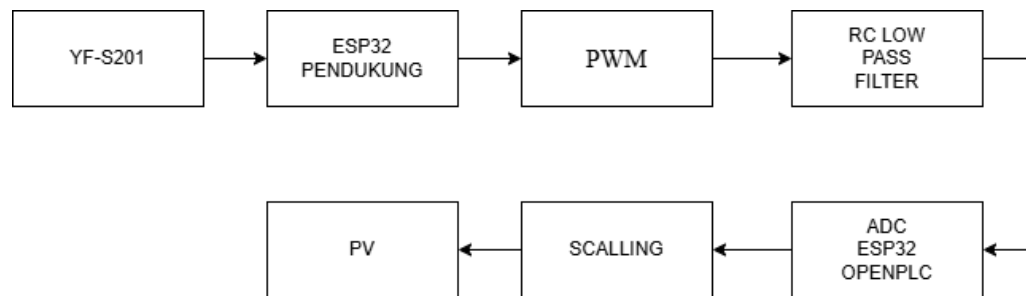
Dengan demikian, komunikasi *pin-to-pin* dapat meneruskan keluaran kendali OpenPLC pada daerah rendah hingga tinggi dan menghasilkan perubahan nilai kendali pompa sesuai dengan perubahan input yang diberikan.

4.3.5 Verifikasi Jalur Data Flow ESP32 Pendukung ke ESP32 OpenPLC

Selain jalur keluaran kendali, komunikasi juga diverifikasi pada arah feedback. Pulsa YF-S201 diproses oleh ESP32 pendukung menjadi nilai flow dan

PWM 12-bit. Sinyal tersebut dilewatkan melalui RC low-pass filter dan dibaca pada ADC_FLOW ESP32 OpenPLC.

Selama pengujian open-loop dan closed-loop, perubahan laju aliran pada YF-S201 diikuti oleh perubahan ADC_FLOW dan FLOW2 pada OpenPLC. Hasil ini menunjukkan bahwa jalur dapat meneruskan perubahan *process variable* menuju CTRL_PID.



Gambar 4.24 Diagram Blok Proses Perubahan Proses Variabel

Dengan demikian, komunikasi antar ESP32 pada sistem bersifat dua arah: jalur feedback membawa data *flow* menuju OpenPLC, sedangkan jalur kendali membawa keluaran CTRL_PID menuju ESP32 pendukung dan aktuator.

4.3.6. Analisis Komunikasi Dua Arah

Setelah dilakukan pengujian menggunakan empat kondisi PWM, seluruh hasil kemudian dibandingkan untuk mengetahui kecenderungan perubahan sinyal pada jalur komunikasi.

Hasil pengujian dirangkum pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian Komunikasi *Pin-to-Pin* OpenPLC menuju ESP32 Pendukung

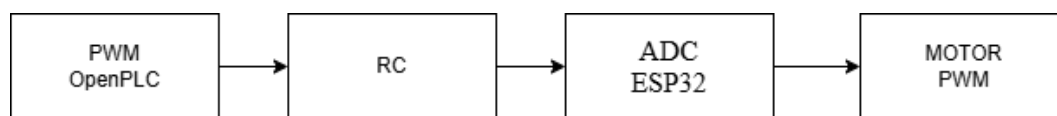
PWM OpenPLC	Tegangan setelah RC Filter (V)	PID Flow ADC	Motor PWM
0	0,0924	0	0
22.000	1,113	1369	1369
32.000	1,594	1877	1877
44.000	2,150	2661	2661

Berdasarkan Tabel 4.4, jalur keluaran kendali menunjukkan hubungan yang konsisten antara PWM OpenPLC, tegangan setelah RC *low-pass* filter, dan ADC yang diterima ESP32 pendukung.

Pada kondisi awal PWM 0, tegangan yang terukur hanya sebesar 0,0924 V dan nilai PID Flow ADC serta Motor PWM berada pada nilai 0. Ketika nilai PWM dinaikkan menjadi 22.000, tegangan meningkat menjadi 1,113 V dan ADC meningkat menjadi 1369.

Selanjutnya, ketika PWM dinaikkan menjadi 32.000, tegangan meningkat menjadi 1,594 V dan nilai ADC menjadi 1877. Pada kondisi PWM 44.000, tegangan kembali meningkat menjadi 2,150 V dan ADC mencapai 2661.

Dengan demikian, diperoleh kecenderungan:



Gambar 4.25 Diagram Blok

Hasil tersebut menunjukkan bahwa jalur OpenPLC → PWM → RC → ADC → ESP32 pendukung mampu meneruskan perubahan keluaran pengendali secara monoton. Arah sebaliknya juga terverifikasi dari perubahan YF-S201 yang diikuti perubahan ADC_FLOW dan FLOW2 pada OpenPLC.

Perbedaan nilai numerik antara PWM OpenPLC dengan PID *Flow* ADC merupakan kondisi yang wajar karena kedua nilai berada pada skala yang berbeda. Keluaran OpenPLC berada pada rentang yang lebih besar, sedangkan ADC ESP32 menggunakan resolusi 12-bit. Selain itu, nilai tersebut telah melalui perubahan bentuk sinyal dari nilai digital menjadi PWM, kemudian menjadi tegangan analog melalui RC filter, dan kembali menjadi nilai digital melalui ADC.

Secara fisik, rangkaian RC *low-pass* filter memiliki peran penting dalam komunikasi ini. Sinyal PWM pada dasarnya berupa perubahan logika HIGH dan LOW dengan frekuensi tertentu. Apabila sinyal tersebut langsung dibaca menggunakan ADC tanpa proses penyaringan, nilai pembacaan dapat mengalami perubahan yang sangat cepat bergantung pada posisi pembacaan terhadap pulsa.

RC low-pass filter meratakan perubahan tersebut sehingga ADC menerima tegangan rata-rata yang lebih stabil dan sesuai dengan duty cycle PWM. Oleh karena itu, semakin tinggi duty cycle yang diberikan, semakin tinggi pula tegangan rata-rata yang diterima ADC.

Meskipun demikian, nilai tegangan aktual tidak selalu sama persis dengan nilai teoritis. Hal ini dapat dipengaruhi oleh toleransi resistor dan kapasitor, tegangan logika ESP32 yang tidak selalu tepat 3,3 V, impedansi input ADC, karakteristik ADC ESP32, panjang dan kualitas kabel, common ground, serta noise pada sistem elektronika.

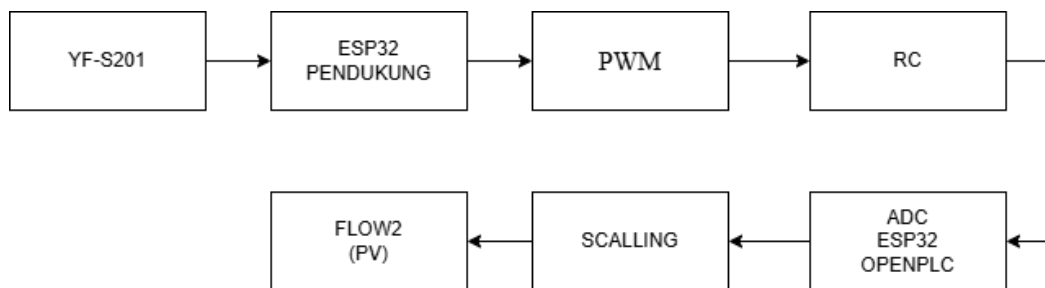
Hal tersebut terlihat pada kondisi PWM 0, ketika masih terdapat tegangan sebesar 0,0924 V. Namun, karena nilainya sangat rendah, program tetap mampu mengenalinya sebagai kondisi minimum sehingga Motor PWM tetap bernilai 0.

Hal penting lainnya adalah jalur komunikasi tersebut menunjukkan karakteristik monoton, yaitu peningkatan nilai pada sisi pengirim selalu diikuti peningkatan nilai pada sisi penerima. Tidak ditemukan kondisi ketika nilai PWM dinaikkan tetapi tegangan atau ADC justru menurun pada empat kondisi uji yang digunakan.

Karakteristik tersebut penting pada sistem closed-loop karena keluaran PID berubah secara dinamis sesuai error antara setpoint dan process variable. Ketika PID memberikan aksi kendali yang lebih besar, nilai yang diterima ESP32 pendukung juga harus meningkat agar pompa dapat meningkatkan laju aliran. Sebaliknya, ketika keluaran PID dikurangi, nilai kendali pompa juga harus mengalami penurunan.

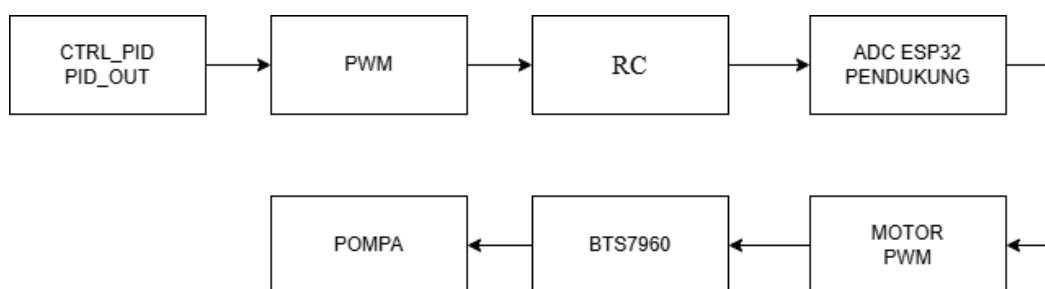
Dengan demikian, komunikasi dua arah pada sistem dapat diringkas menjadi:

Feedback:



Gambar 4.26 Diagram Blok Alu Komunikasi 2 Arah

Kendali:



Gambar 4.27 Diagram Komunikasi 2 Arah

Kedua jalur tersebut telah mampu meneruskan perubahan nilai sesuai arah fungsi masing-masing dan mendukung operasi closed-loop.

Keberhasilan komunikasi pin-to-pin ini juga mendukung hasil pengujian closed-loop pada Subbab 4.6. Pada pengujian tersebut, perubahan keluaran PID mampu mengubah kecepatan pompa sehingga process variable dapat bergerak menuju nilai setpoint. Apabila komunikasi *pin-to-pin* tidak bekerja dengan benar, perubahan keluaran PID dari OpenPLC tidak akan dapat diteruskan menuju pompa sehingga proses koreksi *closed-loop* tidak dapat berlangsung.

Berdasarkan seluruh hasil pengujian, komunikasi pin-to-pin dapat dinyatakan berfungsi untuk meneruskan feedback flow menuju OpenPLC dan keluaran kendali menuju ESP32 pendukung. Perbedaan skala numerik merupakan konsekuensi dari konversi PWM–RC–ADC dan tidak mengubah kecenderungan perubahan sinyal yang diperlukan oleh *loop* kontrol.

Berdasarkan metode pengujian yang dilakukan, jalur kendali dari ESP32 OpenPLC menuju ESP32 pendukung telah diverifikasi secara kuantitatif melalui

hubungan nilai PWM, tegangan keluaran RC *low-pass* filter, pembacaan ADC, dan Motor PWM. Sementara itu, jalur feedback dari ESP32 pendukung menuju ESP32 OpenPLC diverifikasi secara fungsional melalui perubahan ADC_FLOW dan FLOW2 ketika laju aliran berubah. Dengan demikian, hasil pengujian menunjukkan bahwa kedua jalur dapat mendukung operasi *closed-loop*, tetapi pengujian ini belum digunakan untuk menentukan error transfer numerik komunikasi pada kedua arah secara terpisah.

4.4. Pengujian Open-Loop Sistem Pengendalian Laju Aliran

Pengujian open-loop dilakukan untuk mengetahui karakteristik dasar plant sebelum *Function Block* CTRL_PID digunakan pada sistem pengendalian. Pada pengujian ini, nilai *Pulse Width Modulation* (PWM) diberikan secara langsung dan dipertahankan konstan selama proses pengambilan data. Nilai *process variable* yang diperoleh dari sensor tidak digunakan untuk melakukan koreksi terhadap nilai PWM, sehingga perubahan laju aliran yang terjadi merupakan respons alami plant terhadap masukan yang diberikan.

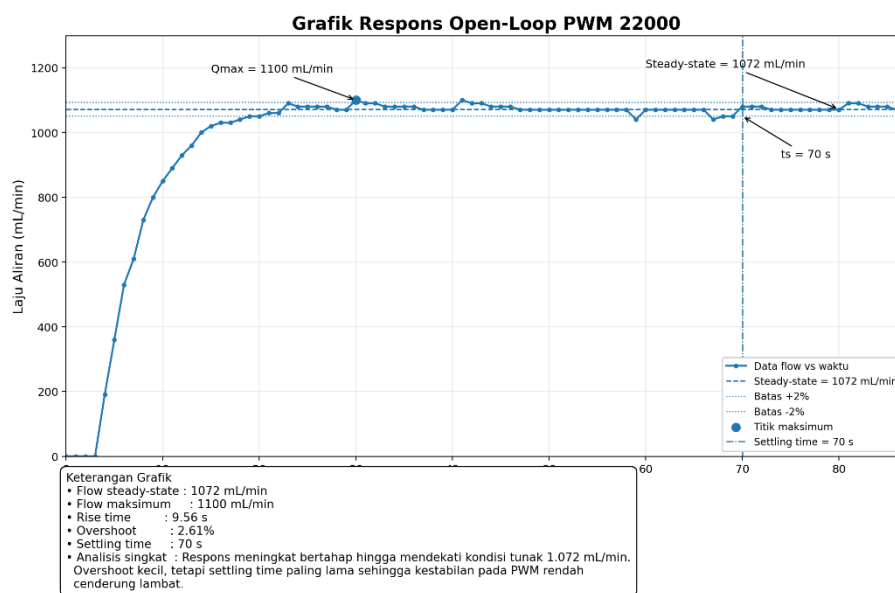
Pengujian dilakukan menggunakan tiga variasi nilai PWM, yaitu 22.000, 30.000, dan 44.000. Ketiga nilai tersebut dipilih untuk mewakili daerah operasi laju aliran rendah, menengah, dan tinggi. Pada setiap pengujian, perubahan process variable laju aliran direkam terhadap waktu mulai dari kondisi awal hingga respons mencapai kondisi yang relatif stabil.

Parameter yang dianalisis pada pengujian open-loop meliputi nilai kondisi tunak (Q_{ss}), nilai maksimum respons (Q_{peak}), *rise time*, *maximum overshoot*, dan *settling time*. Penentuan settling time dilakukan menggunakan batas toleransi sebesar $\pm 2\%$ terhadap nilai kondisi tunak sesuai dengan metode pengolahan data yang digunakan pada grafik respons.

Pengujian open-loop juga digunakan untuk menentukan titik kerja yang digunakan dalam proses identifikasi *process reaction curve*. Berdasarkan karakteristik ketiga titik pengujian, PWM 30.000 dipilih sebagai titik kerja identifikasi karena berada pada daerah operasi menengah dan menghasilkan kurva respons yang dapat diamati dengan baik.

4.4.1. Pengujian *Open-Loop* PWM 22.000

Pengujian pertama dilakukan dengan memberikan nilai PWM sebesar 22.000 secara konstan kepada pompa. Nilai tersebut digunakan untuk mengetahui karakteristik respons plant pada daerah operasi laju aliran rendah. Setelah PWM diberikan, perubahan laju aliran direkam sejak kondisi awal hingga respons mencapai kondisi yang relatif stabil.



Gambar 4.28 Respons Laju Aliran Open-Loop pada PWM 22.000

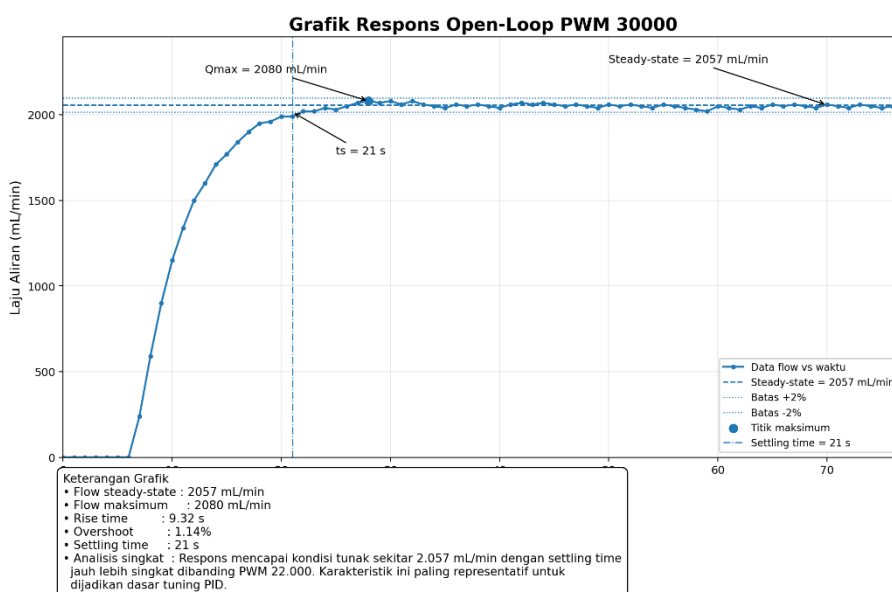
Berdasarkan Gambar 4.23, pemberian PWM sebesar 22.000 menghasilkan respons laju aliran yang meningkat secara bertahap dari kondisi awal hingga mencapai kondisi tunak sekitar 1.072 mL/min. Nilai maksimum respons yang tercatat sebesar 1.100 mL/min. Berdasarkan metode pengolahan data yang telah dijelaskan pada Bab III, diperoleh rise time sebesar 9,56 s, maximum overshoot sebesar 2,61%, dan settling time sebesar 70 s dengan batas toleransi $\pm 2\%$.

Nilai *rise time* sebesar 9,56 s menunjukkan bahwa plant memerlukan waktu sekitar 9,56 s untuk bergerak dari 10% hingga 90% nilai kondisi tunaknya. Sementara itu, *overshoot* sebesar 2,61% menunjukkan bahwa lonjakan respons setelah mencapai daerah kondisi tunak relatif kecil. Meskipun demikian, *settling time* sebesar 70 s merupakan yang paling lama dibandingkan dua kondisi PWM lainnya. Hal tersebut disebabkan masih terdapat beberapa fluktuasi pembacaan di

sekitar nilai kondisi tunak yang menyebabkan respons keluar kembali dari batas toleransi $\pm 2\%$. Dengan demikian, pada daerah operasi rendah sistem dapat mencapai kondisi tunak, tetapi membutuhkan waktu yang lebih lama untuk benar-benar stabil pada daerah toleransi yang ditentukan.

4.4.2. Pengujian *Open-Loop* PWM 30.000

Pengujian kedua dilakukan dengan memberikan nilai PWM sebesar 30.000 secara konstan kepada pompa. Nilai PWM tersebut digunakan untuk merepresentasikan daerah operasi menengah plant. Prosedur pengujian dilakukan dengan cara yang sama seperti pada PWM 22.000, yaitu dengan merekam perubahan laju aliran mulai dari kondisi awal hingga mencapai kondisi yang relatif stabil.



Gambar 4.29 Respons Laju Aliran *Open-Loop* pada PWM 30.000

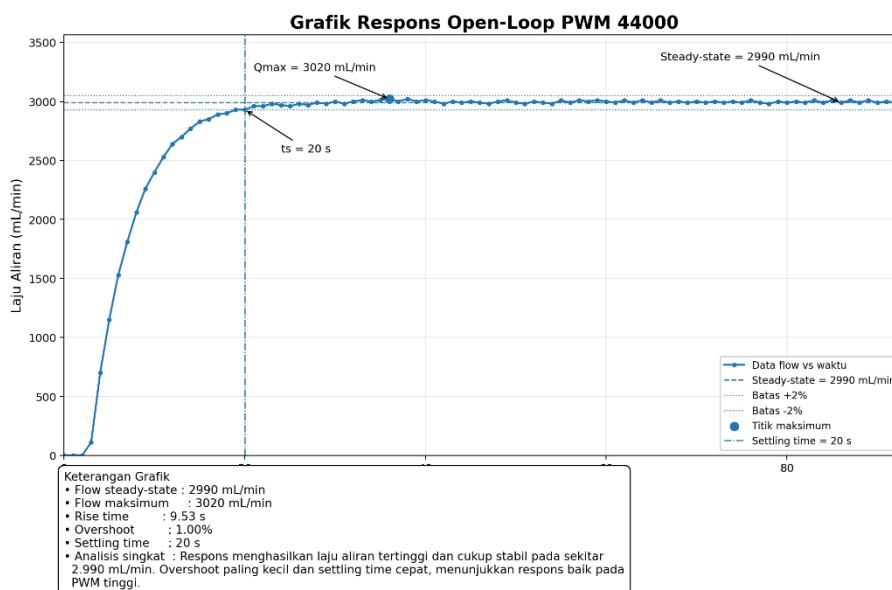
Berdasarkan Gambar 4.24, pemberian PWM sebesar 30.000 menghasilkan kenaikan laju aliran dari kondisi awal hingga mencapai nilai kondisi tunak sekitar 2.057 mL/min. Nilai maksimum respons yang tercatat sebesar 2.080 mL/min. Hasil pengolahan data menunjukkan *rise time* sebesar 9,32 s, *maximum overshoot* sebesar 1,14%, dan *settling time* sebesar 21 s dengan batas toleransi $\pm 2\%$.

Dibandingkan PWM 22.000, respons pada PWM 30.000 memiliki *rise time* yang hampir sama, tetapi mampu mencapai kondisi stabil dalam waktu yang jauh

lebih singkat. Nilai *overshoot* sebesar 1,14% menunjukkan bahwa respons hanya mengalami kenaikan kecil di atas nilai kondisi tunaknya. Setelah sekitar 21 s, respons berada di dalam daerah toleransi $\pm 2\%$ dan cenderung tetap stabil. Karakteristik tersebut menunjukkan bahwa titik kerja PWM 30.000 memiliki respons yang cukup baik dan representatif untuk menggambarkan karakteristik plant pada daerah operasi menengah. Oleh karena itu, data pada PWM 30.000 selanjutnya digunakan sebagai dasar penentuan parameter *process reaction curve* untuk memperoleh parameter awal PID menggunakan metode Ziegler–Nichols open loop.

4.4.3. Pengujian *Open-Loop* PWM 44.000

Pengujian ketiga dilakukan dengan memberikan nilai PWM sebesar 44.000 secara konstan kepada pompa. Pengujian ini digunakan untuk mengetahui karakteristik plant pada daerah operasi laju aliran tinggi. Nilai laju aliran kembali direkam sejak kondisi awal hingga respons mencapai kondisi yang relatif stabil.



Gambar 4.30 Respons Laju Aliran *Open-Loop* pada PWM 44.000

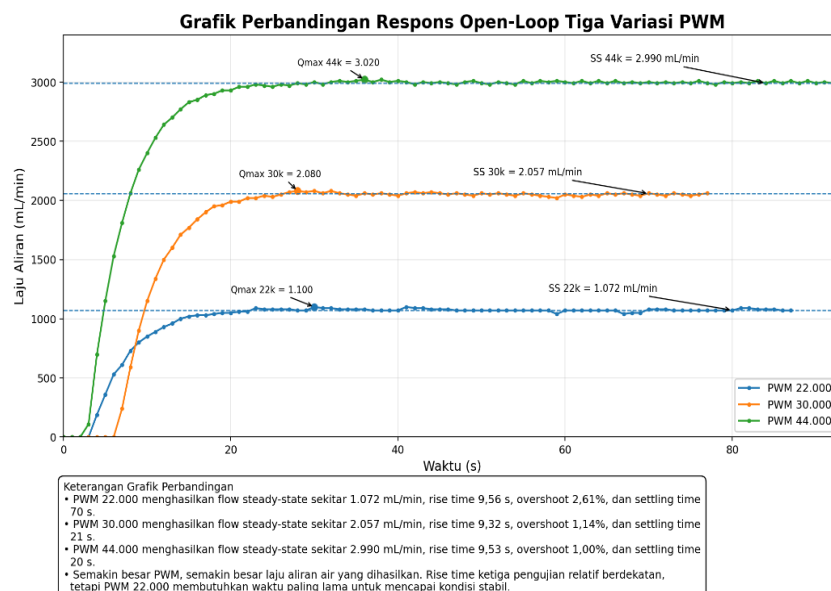
Berdasarkan Gambar 4.25, pemberian PWM sebesar 44.000 menghasilkan kondisi tunak laju aliran sekitar 2.990 mL/min, sedangkan nilai maksimum respons yang tercatat sebesar 3.020 mL/min. Hasil analisis menunjukkan *rise time* sebesar

9,53 s, *maximum overshoot* sebesar 1,00%, dan *settling time* sebesar 20 s dengan batas toleransi $\pm 2\%$.

Nilai *rise time* pada PWM 44.000 tidak berbeda jauh dibandingkan pengujian PWM 22.000 dan PWM 30.000. Hal tersebut menunjukkan bahwa perubahan besar PWM lebih dominan memengaruhi besar laju aliran kondisi tunak dibandingkan kecepatan awal respons plant. Nilai overshoot sebesar 1,00% merupakan yang paling kecil dari ketiga pengujian dan menunjukkan bahwa respons pada daerah operasi tinggi tidak mengalami lonjakan yang besar. Settling time sebesar 20 s juga menunjukkan bahwa respons mampu masuk dan bertahan pada daerah toleransi relatif cepat. Meskipun demikian, kondisi ini berada lebih dekat dengan batas atas kemampuan operasi sistem sehingga pada pengujian closed-loop perlu diperhatikan kemungkinan terjadinya saturasi keluaran.

4.4.4. Perbandingan Hasil Pengujian *Open-Loop*

Setelah dilakukan pengujian menggunakan tiga nilai PWM, respons masing-masing kondisi dibandingkan untuk mengetahui perubahan karakteristik plant akibat peningkatan sinyal kendali. Grafik perbandingan ketiga respons ditunjukkan pada Gambar 4.26.



Gambar 4.31 Perbandingan Respons *Open-Loop* PWM 22.000, 30.000, dan 44.000

Tabel 4.5. Perbandingan Parameter Respons Sistem *Open-Loop*

PWM	Qss (mL/min)	Peak (mL/min)	Rise Time (s)	Overshoot (%)	Settling Time $\pm 2\%$ (s)
22.000	1.072	1.100	9,56	2,61	70
30.000	2.057	2.080	9,32	1,14	21
44.000	2.990	3.020	9,53	1,00	20

Berdasarkan Gambar 4.26 dan Tabel 4.5, peningkatan nilai PWM menghasilkan peningkatan nilai laju aliran kondisi tunak. PWM 22.000 menghasilkan kondisi tunak sekitar 1.072 mL/min, PWM 30.000 menghasilkan sekitar 2.057 mL/min, sedangkan PWM 44.000 menghasilkan sekitar 2.990 mL/min. Hasil tersebut menunjukkan bahwa besar sinyal PWM memiliki hubungan yang jelas terhadap besar laju aliran yang dihasilkan oleh plant.

Nilai *rise time* pada ketiga pengujian berada pada rentang yang relatif berdekatan, yaitu 9,32–9,56 s. Hal ini menunjukkan bahwa perubahan PWM dari 22.000 hingga 44.000 tidak menghasilkan perbedaan yang besar terhadap waktu kenaikan awal respons. Perbedaan yang lebih jelas terlihat pada *settling time*. Pada PWM 22.000 diperoleh *settling time* sebesar 70 s, sedangkan pada PWM 30.000 dan 44.000 masing-masing hanya sebesar 21 s dan 20 s. Dengan demikian, respons pada daerah operasi rendah membutuhkan waktu yang lebih lama untuk bertahan di dalam daerah toleransi $\pm 2\%$.

Nilai *maximum overshoot* ketiga kondisi tergolong relatif kecil, yaitu 2,61% pada PWM 22.000, 1,14% pada PWM 30.000, dan 1,00% pada PWM 44.000. Kecilnya overshoot menunjukkan bahwa *plant* tidak menunjukkan respons transien yang sangat berlebihan ketika diberikan nilai PWM tetap.

Berdasarkan keseluruhan hasil pengujian, PWM 30.000 dipilih sebagai titik kerja untuk penentuan parameter *process reaction curve*. Pemilihan tersebut dilakukan karena PWM 30.000 berada pada daerah operasi menengah, menghasilkan respons yang relatif stabil, memiliki overshoot yang kecil, serta tidak berada terlalu dekat dengan kondisi operasi minimum maupun maksimum sistem.

4.5. Pengujian dan Analisa Kontrol PID

Setelah karakteristik dasar *plant* diperoleh melalui pengujian open-loop, tahap selanjutnya adalah mengimplementasikan kontrol *Proportional-Integral-Derivative* (PID) pada sistem pengendalian laju aliran air. Pengendali PID diimplementasikan menggunakan *Function Block* CTRL_PID yang dijalankan oleh OpenPLC Runtime pada ESP32.

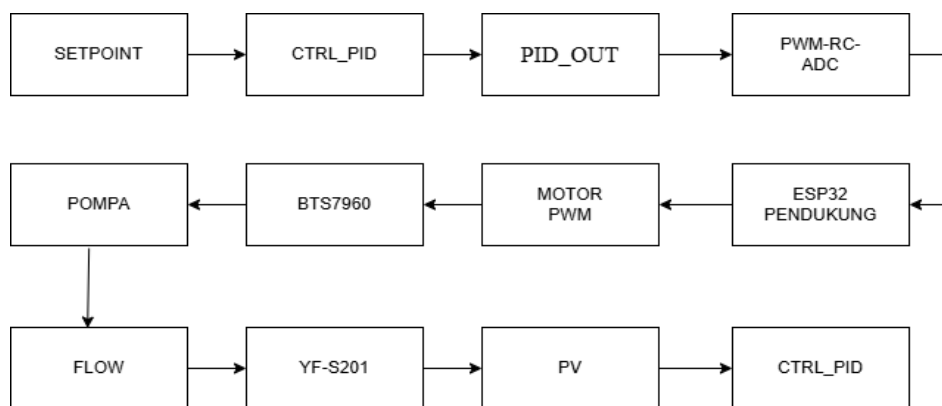
Parameter awal pengendali diperoleh menggunakan metode *Ziegler–Nichols open loop* berdasarkan *process reaction curve* pada pengujian PWM 30.000. Parameter tersebut selanjutnya diterapkan pada *Function Block* CTRL_PID sebagai parameter awal sebelum dilakukan penyesuaian atau *fine tuning* berdasarkan respons aktual *plant*.

Pada sistem *closed-loop*, nilai laju aliran hasil pengolahan sensor digunakan sebagai *process variable* (PV), sedangkan nilai laju aliran yang diinginkan digunakan sebagai *setpoint* (SP). OpenPLC membandingkan kedua nilai tersebut dan menghitung error menggunakan:

Hubungan error antara *setpoint* dan *process variable* mengacu pada Persamaan PID pada Subbab 2.6.4.

CTRL_PID memproses perbedaan antara SET dan ACT berdasarkan parameter KP, TN, dan TV. Keluaran pengendali disimpan sebagai PID_OUT dan dikirim melalui jalur PWM–RC–ADC menuju ESP32 pendukung sebelum diterapkan sebagai Motor PWM pada BTS7960.

Secara umum, proses pengendalian closed-loop dapat dinyatakan sebagai:



Gambar 4.32 Diagram Blok Hasil Proses Pengendalian Closed Loop

Pada pengujian akhir, sistem menggunakan tiga nilai setpoint secara langsung, yaitu:

$$SP1 = 1000 \text{ mL/min}$$

$$SP2 = 2000 \text{ mL/min}$$

$$SP3 = 3000 \text{ mL/min}$$

Ketiga nilai tersebut digunakan untuk mengevaluasi kemampuan pengendali pada daerah operasi laju aliran rendah, menengah, dan tinggi.

4.5.1. Penentuan Parameter Awal PID Metode Ziegler–Nichols Open-Loop

Parameter awal pengendali PID dihitung menggunakan metode Ziegler–Nichols open loop berdasarkan parameter *process reaction curve*. Parameter yang digunakan dalam perhitungan terdiri atas perubahan masukan (Δu), dead time (L), dan kemiringan maksimum respons (R).

Berdasarkan hasil identifikasi respons open-loop pada PWM 30.000 diperoleh parameter:

$$L = 6,104 \text{ s}$$

$$R = 247,714 \text{ mL/min/s}$$

Dan perubahan masukan:

$$\Delta u = 30.000$$

Parameter tersebut kemudian digunakan untuk menentukan nilai awal pengendali PID.

Nilai tersebut digunakan sebagai initial tuning. Selanjutnya dilakukan fine tuning secara eksperimental karena respons sistem aktual masih memerlukan penyesuaian terhadap karakteristik plant.

Parameter hasil *Ziegler–Nichols* digunakan sebagai initial tuning dan tidak langsung ditetapkan sebagai parameter akhir. Setelah parameter awal diterapkan, dilakukan penyesuaian eksperimental untuk memperoleh respons yang lebih sesuai dengan karakteristik plant.

Parameter tersebut digunakan sebagai initial tuning untuk pengujian closed-loop. Hasil pengujian awal menunjukkan bahwa respons aktual masih memerlukan penyesuaian, sehingga dilakukan *fine tuning* secara eksperimental sebelum pengujian akhir.

Tabel 4.6. Parameter Awal Hasil Metode Ziegler–Nichols Open-Loop

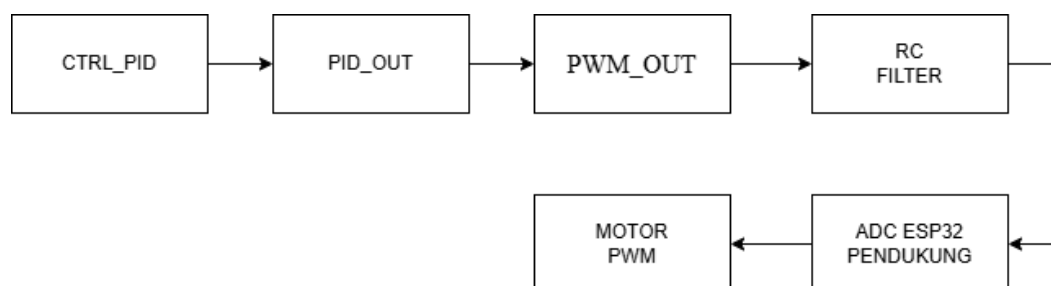
Parameter CTRL_PID	Nilai
KP	23,809
Ki	1,950
Kd	72,665

4.5.2. Proses Pembentukan Keluaran PID menjadi PWM

Keluaran *Function Block* CTRL_PID berupa variabel Y yang disimpan sebagai PID_OUT. Nilai tersebut dibatasi oleh parameter LL = 0 dan LH = 65.535. PID_OUT selanjutnya digunakan untuk membentuk sinyal PWM pada sisi ESP32 OpenPLC. Sinyal PWM tersebut tidak dikirim sebagai nilai numerik digital secara langsung, tetapi dilewatkan melalui RC *low-pass* filter sehingga menghasilkan tegangan rata-rata. Tegangan tersebut kemudian dibaca oleh ADC 12-bit pada ESP32 pendukung.

Nilai ADC yang diterima berada pada rentang 0–4095 dan selanjutnya digunakan sebagai Motor PWM pada driver BTS7960. Oleh karena itu, nilai PID_OUT dan Motor PWM tidak memiliki nilai numerik yang sama secara langsung karena telah melalui proses konversi PWM–RC–ADC.

Nilai tersebut kemudian digunakan sebagai duty cycle PWM yang diberikan kepada driver BTS7960. Secara umum, proses pembentukan keluaran dapat dituliskan sebagai:

**Gambar 4.33 Diagram Blok Proses PID OUT Ke Pompa**

Semakin besar nilai keluaran pengendali, semakin besar *duty cycle* yang diterapkan pada pompa. Sebaliknya, ketika nilai *process variable* mendekati atau

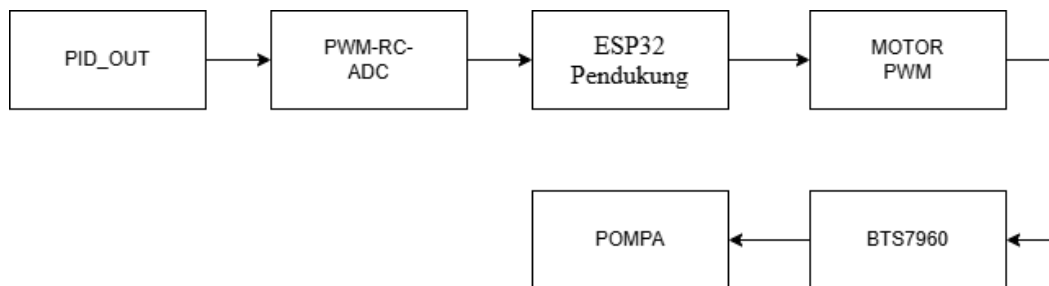
melebihi nilai *setpoint*, keluaran pengendali akan disesuaikan oleh CTRL_PID sesuai dengan nilai error yang terjadi.

Proses konversi ini berfungsi sebagai penghubung antara hasil perhitungan algoritma PID pada OpenPLC dengan sinyal fisik yang digunakan untuk mengatur kecepatan pompa.

4.5.3. Integrasi PID dengan Pompa

Integrasi pengendali PID dengan pompa dilakukan melalui ESP32 pendukung dan driver BTS7960. ESP32 menerima nilai keluaran kendali dari OpenPLC, kemudian nilai tersebut diterapkan sebagai motorPWM.

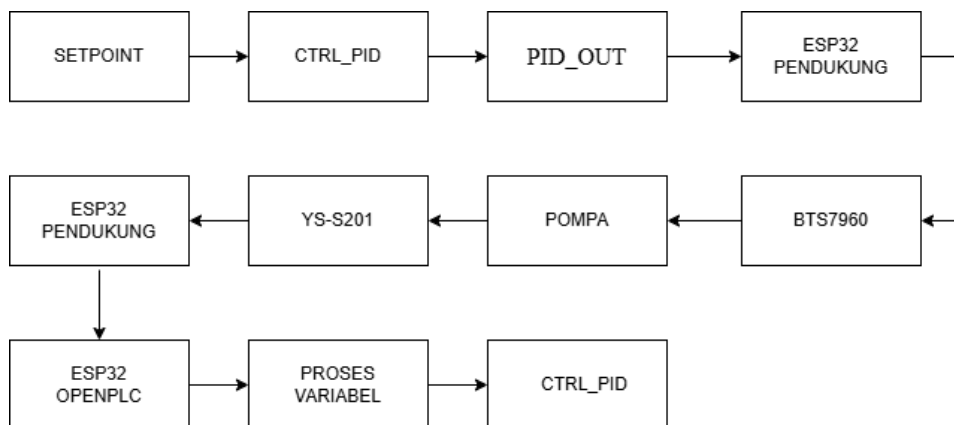
Pada driver BTS7960, sinyal PWM diberikan pada kanal RPWM, sedangkan kanal LPWM dipertahankan pada nilai nol karena pompa hanya digunakan untuk mengalirkan air dalam satu arah. Secara umum, jalur aktuasi sistem dapat dituliskan sebagai:



Gambar 4.34 Diagram Blok Jalur Aktuasi Sistem

Perubahan PWM menyebabkan perubahan kecepatan putaran pompa. Perubahan kecepatan tersebut kemudian mengubah laju aliran air yang melewati sensor YF-S201. Nilai laju aliran kembali dibaca dan digunakan sebagai process variable oleh CTRL_PID sehingga terbentuk sistem closed-loop.

Jalur keseluruhan pengendalian dapat dinyatakan sebagai:



Gambar 4.35 Blok Diagram Pengendalian Sistem

Dengan integrasi tersebut, proses pengaturan kecepatan pompa tidak lagi dilakukan menggunakan nilai PWM tetap seperti pada pengujian open-loop, melainkan berubah secara otomatis berdasarkan error antara setpoint dan process variable.

4.5.4. Verifikasi Implementasi PID pada OpenPLC

Verifikasi implementasi PID dilakukan untuk memastikan bahwa *Function Block* CTRL_PID dapat menerima nilai *process variable* dan *setpoint*, memproses error, serta menghasilkan keluaran kendali yang berubah sesuai kondisi sistem. Pengujian dilakukan dengan memberikan nilai setpoint kemudian mengamati perubahan tiga variabel utama, yaitu:

SP, PV, Y

Dengan:

SP : setpoint laju aliran,

PV : process variable laju aliran,

Y : keluaran Function Block CTRL_PID.

Ketika nilai process variable berada di bawah *setpoint*, *Function Block* menghasilkan aksi kendali untuk meningkatkan keluaran pompa. Ketika process variable mendekati *setpoint*, keluaran kendali mengalami penyesuaian untuk mempertahankan kondisi operasi di sekitar nilai referensi.

Hasil verifikasi menunjukkan bahwa perubahan nilai *setpoint* dan *process variable* dapat diproses oleh CTRL_PID dan menghasilkan perubahan keluaran

PID_OUT. Keluaran tersebut dapat diteruskan menuju sistem aktuator sehingga membentuk pengendalian laju aliran secara *closed-loop*.

Dengan demikian, implementasi CTRL_PID pada OpenPLC telah berhasil diintegrasikan dengan ESP32 dan pompa sebagai bagian utama sistem pengendalian laju aliran air.

4.6. Pengujian Sistem Menggunakan PID

Pengujian sistem menggunakan PID dilakukan untuk mengetahui kemampuan pengendali dalam mengatur laju aliran air agar *process variable* mengikuti nilai *setpoint* yang diberikan. Pengujian dilakukan secara *closed-loop* menggunakan *Function Block* CTRL_PID pada OpenPLC.

Sebelum pengujian akhir dilakukan *fine tuning* dengan membandingkan dua konfigurasi parameter PID, yaitu $K_p = 5$, $K_i = 5$, dan $K_d = 0,5$, serta $K_p = 3,5$, $K_i = 3,5$, dan $K_d = 0,35$. Berdasarkan hasil perbandingan, konfigurasi $K_p = 3,5$, $K_i = 3,5$, dan $K_d = 0,35$ dipilih untuk pengujian akhir pada setpoint 1000, 2000, dan 3000 mL/min. Berdasarkan hasil perbandingan, konfigurasi $K_p = 3,5$, $K_i = 3,5$, $K_d = 0,35$ dipilih untuk pengujian akhir pada setpoint 1000, 2000, dan 3000 mL/min.

Parameter respons yang dianalisis meliputi *rise time*, *maximum overshoot*, rata-rata *process variable*, bias terhadap *setpoint*, standar deviasi, koefisien variasi, MAE, RMSE, serta persentase data dalam batas $\pm 5\%$ dan $\pm 10\%$. Metode pengolahan data mengacu pada Subbab 3.6.5.

4.6.1. Hasil *Fine Tuning* Parameter PID

Fine tuning dilakukan setelah parameter awal hasil metode Ziegler–Nichols diterapkan pada plant aktual. *Fine tuning* dilakukan setelah parameter awal hasil metode Ziegler–Nichols diterapkan pada plant aktual. Parameter awal sebesar $K_p = 23,809$, $K_i = 1,950$, dan $K_d = 72,665$ belum langsung digunakan sebagai parameter akhir karena respons aktual sistem masih memerlukan penyesuaian

Pada tahap evaluasi *fine tuning* yang didokumentasikan dalam penelitian ini digunakan dua konfigurasi parameter, yaitu:

Konfigurasi A: $K_p = 5$, $K_i = 5$, $K_d = 0,5$

Dan

Konfigurasi B: $K_p = 3,5$, $K_i = 3,5$, $K_d = 0,35$

Kedua konfigurasi tersebut dibandingkan berdasarkan respons yang diperoleh pada tiga *setpoint* pengujian. Karena data perubahan parameter secara bertahap dari parameter awal *Ziegler–Nichols* menuju dua konfigurasi tersebut belum disajikan sebagai data eksperimen tersendiri, pembahasan kuantitatif pada bagian ini difokuskan pada dua konfigurasi yang memiliki hasil pengujian terdokumentasi.

Hasil perbandingan *maximum overshoot* kedua parameter ditunjukkan pada Tabel 4.7.

Tabel 4.7 Perbandingan *Overshoot* Hasil *Fine Tuning*

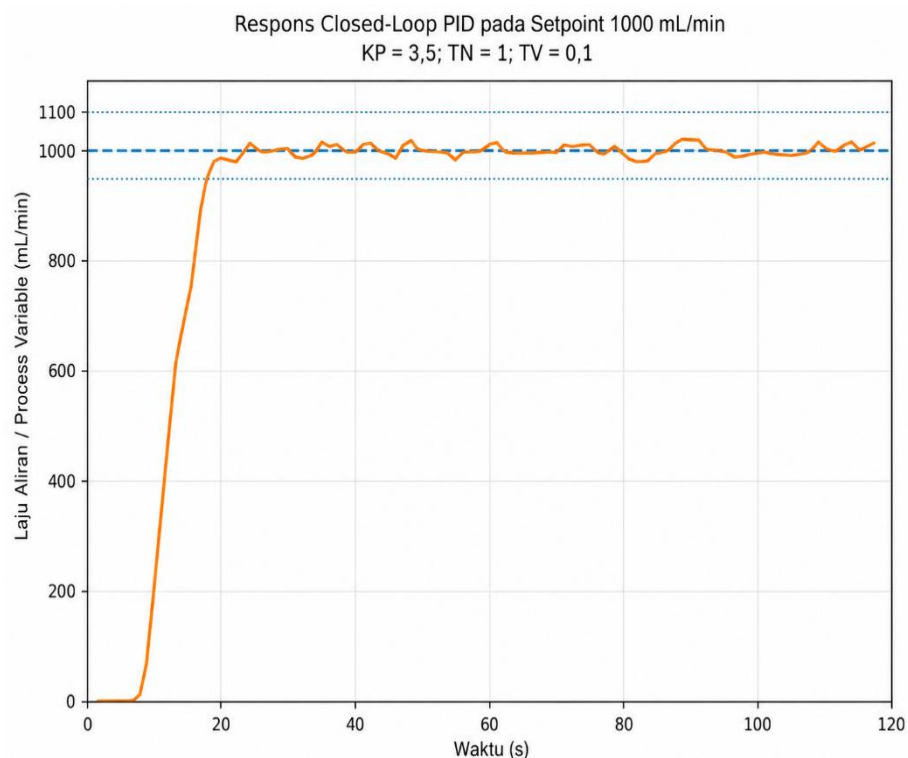
Setpoint (mL/min)	Overshoot Konfigurasi A (%)	Overshoot Konfigurasi B (%)
1.000	245,94	9,36
2.000	74,19	74,96
3.000	16,65	16,67

Berdasarkan Tabel 4.7, penurunan K_p dari 5 menjadi 3,5 memberikan pengaruh paling signifikan pada *setpoint* 1000 mL/min, yaitu menurunkan *maximum overshoot* tercatat dari 245,94% menjadi 9,36%. Pada *setpoint* 2000 mL/min, nilai *overshoot* berubah dari 74,19% menjadi 74,96%, sedangkan pada *setpoint* 3000 mL/min berubah dari 16,65% menjadi 16,67%. Dengan demikian, penurunan K_p tidak memberikan penurunan *overshoot* pada seluruh titik operasi.

Parameter $K_p=3,5$, $K_i=3,5$, $K_d=0,35$ selanjutnya digunakan sebagai konfigurasi akhir karena mampu mengurangi respons yang sangat agresif pada daerah *flow* rendah, sedangkan evaluasi lengkap terhadap performa konfigurasi tersebut pada masing-masing titik operasi dilakukan berdasarkan hasil *closed-loop* pada Subbab 4.6.2 sampai 4.6.5.

4.6.2. Pengujian *Closed-Loop Setpoint* 1000 mL/min

Pengujian *closed-loop* pertama dilakukan pada *setpoint* 1000 mL/min menggunakan parameter akhir $K_p = 3,5$, $K_i = 3,5$, dan $K_d = 0,35$. Pengujian digunakan untuk mengevaluasi kemampuan pengendali pada daerah operasi laju aliran rendah.



Gambar 4.36 Respons *Closed-Loop* pada *Setpoint* 1000 mL/min

Berdasarkan Gambar 4.28, keluaran sistem mulai memberikan respons setelah pompa diaktifkan dan *process variable* selanjutnya meningkat menuju nilai setpoint. Dari data asli diperoleh nilai maksimum *process variable* sebesar 1093,59 mL/min, rise time sebesar 2,051 s, dan simpangan maksimum terhadap *setpoint* sebesar 9,36%. Pada rentang analisis kondisi tunak 30–90 s, diperoleh rata-rata *process variable* sebesar 1001,11 mL/min, sehingga selisih rata-rata terhadap setpoint hanya sekitar 1,11 mL/min atau 0,111%. Nilai MAE sebesar 36,17 mL/min atau 3,617%, sedangkan RMSE sebesar 48,75 mL/min atau 4,875%.

Apabila diamati lebih rinci, *process variable* belum langsung berubah pada saat keluaran PWM mulai diberikan. Pada data pengujian, keluaran PWM mulai

meningkat sekitar detik ke-8, sedangkan perubahan *process variable* mulai terlihat sekitar detik ke-12. Keterlambatan beberapa detik tersebut tidak hanya berasal dari proses komputasi, tetapi merupakan gabungan dari karakteristik fisik sistem dan proses pengukuran. Setelah pompa mulai berputar, diperlukan waktu untuk membentuk aliran di dalam pipa, menggerakkan rotor sensor YF-S201, menghasilkan pulsa, melakukan penghitungan pulsa pada ESP32, mengirimkan sinyal menuju OpenPLC, dan melewati data melalui low-pass filter sebelum nilai tersebut digunakan sebagai *process variable*.

Fluktuasi yang masih terlihat setelah *process variable* mencapai setpoint juga dapat dijelaskan dari karakteristik pengukuran YF-S201. Sensor tersebut menghasilkan pulsa berdasarkan putaran rotor dan menggunakan faktor nominal sekitar 7,5 pulsa per detik untuk setiap 1 L/min. Dengan interval penghitungan sekitar satu detik, pada laju aliran 1 L/min hanya tersedia sekitar 7–8 pulsa dalam satu periode pengukuran. Perbedaan satu pulsa saja secara teoritis dapat merepresentasikan perubahan sekitar 0,133 L/min atau 133 mL/min. Oleh karena itu, pada daerah flow rendah, perubahan satu pulsa mempunyai pengaruh relatif yang cukup besar terhadap nilai flow yang dihitung. Penelitian mengenai YF-S201 juga menunjukkan pentingnya kalibrasi sensor karena nilai konstanta sensor dapat memengaruhi ketelitian pembacaan flow.

Kondisi tersebut sesuai dengan data steady-state SP 1000 mL/min, di mana *process variable* sesaat masih bergerak dari sekitar 881,79 mL/min sampai 1093,59 mL/min, walaupun rata-ratanya sangat dekat dengan *setpoint*. Pada saat yang sama, nilai PWM pada rentang *steady-state* hanya berubah pada daerah sekitar 23.645 sampai 24.524, sehingga tidak terdapat indikasi saturasi keluaran seperti yang terjadi pada setpoint lebih tinggi. Ketika *process variable* turun, keluaran PWM cenderung dinaikkan, sedangkan ketika *process variable* naik, PWM dikurangi. Pola tersebut merupakan karakteristik normal dari sistem umpan balik negatif yang sedang berusaha mengoreksi error.

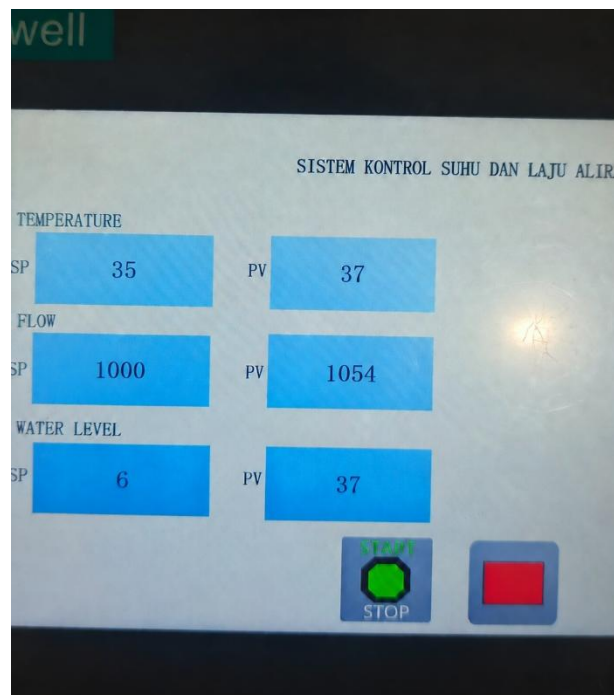
Hal lain yang perlu diperhatikan adalah nilai maksimum 1093,59 mL/min terjadi setelah sistem telah berada di sekitar setpoint, bukan sebagai lonjakan sangat besar pada saat awal kenaikan. Oleh sebab itu, nilai 9,36% pada pengujian ini lebih

tepat dipahami sebagai simpangan positif maksimum *process variable* selama periode pengamatan, bukan semata-mata sebagai *overshoot* transien awal seperti pada sistem orde dua ideal. Istilah *maximum overshoot* tetap dapat digunakan agar konsisten dengan metode pada Subbab 3.6.5, tetapi interpretasinya perlu mempertimbangkan fluktuasi pengukuran. Nilai 9,36% pada pengujian ini merupakan simpangan positif maksimum *process variable* terhadap setpoint selama periode pengamatan. Karena nilai maksimum terjadi ketika sistem telah berada di sekitar daerah setpoint, interpretasi nilai tersebut tidak hanya merepresentasikan *overshoot* transien awal, tetapi juga dipengaruhi oleh fluktuasi *process variable* selama pengendalian *closed-loop*.

Sebanyak 70,59% data pada rentang *steady-state* berada di dalam batas $\pm 5\%$ dan 96,08% berada di dalam batas $\pm 10\%$. Dengan toleransi $\pm 10\%$, respons dapat dianggap menetap sekitar detik ke-75, sedangkan pada batas $\pm 5\%$ masih terdapat beberapa titik yang keluar dari daerah toleransi. Hal tersebut menunjukkan bahwa performa rata-rata pengendali sudah sangat baik, tetapi kestabilan sesaat masih dipengaruhi oleh variasi pembacaan sensor dan dinamika aliran.

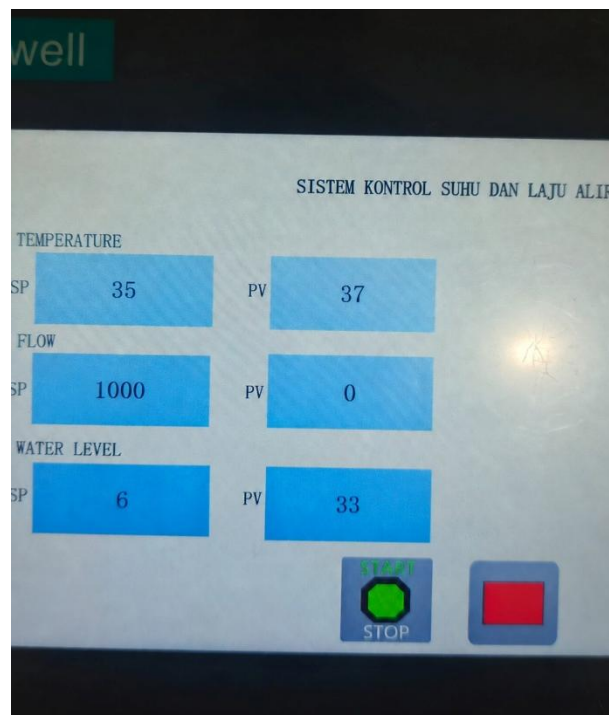
Penggunaan filter yang lebih kuat secara teoritis dapat menghasilkan grafik *process variable* yang lebih halus. Namun, Romero Segovia, Hägglund, dan Åström menjelaskan bahwa terdapat kompromi antara peredaman noise dan performa kontrol: peningkatan *filtering* dapat mengurangi *noise* pengukuran tetapi sekaligus menambah keterlambatan respons. Oleh karena itu, pada sistem ini *low-pass* filter yang telah digunakan sebaiknya tidak diperbesar hanya untuk menghasilkan grafik yang terlihat halus tanpa mempertimbangkan pengaruhnya terhadap dinamika *closed-loop*.

Berdasarkan keseluruhan hasil tersebut, pengendali pada SP 1000 mL/min mampu mempertahankan nilai rata-rata *process variable* sangat dekat dengan *setpoint*. Ketidakhalsan data sesaat lebih banyak menunjukkan kombinasi karakteristik sensor berbasis pulsa, proses sampling, dinamika aliran, dan aksi koreksi *closed-loop* daripada menunjukkan kegagalan algoritma PID.



Gambar 4.37 Tampilan HMI pada Pengujian *Closed-Loop Setpoint* 1000 mL/min

Gambar 4.29 menunjukkan kondisi sistem yang terpantau melalui HMI pada saat pengujian *closed-loop setpoint flow* 1000 mL/min masih berlangsung. Pada kondisi tersebut, nilai process variable flow yang tampil sebesar 1054 mL/min, sehingga laju aliran aktual telah berada di sekitar nilai *setpoint*. Pada bagian water level, *setpoint* diberikan sebesar 6 cm, sedangkan nilai PV yang tampil sebesar 37 cm. Nilai PV pada bagian water level tersebut merupakan jarak antara sensor JSN-SR04T dan permukaan air, bukan tinggi air secara langsung. Oleh karena itu, nilai tersebut digunakan untuk menggambarkan kondisi permukaan air saat proses pengisian masih berlangsung. Tampilan HMI ini menunjukkan kondisi sistem secara real-time ketika pengendalian flow masih berjalan

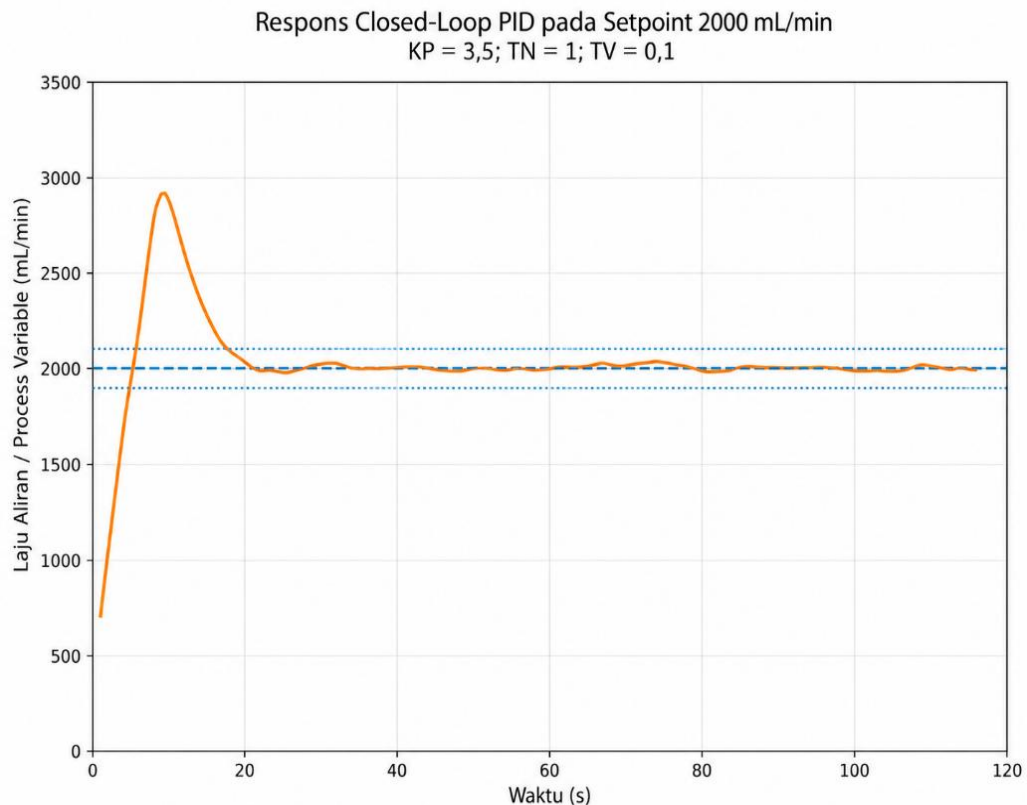


Gambar 4.38 Tampilan HMI Setelah Proses Pengisian pada Setpoint 1000 mL/min

Gambar 4.30 menunjukkan kondisi HMI setelah proses pengisian pada setpoint flow 1000 mL/min dihentikan oleh interlock level. Setelah pompa berhenti, *process variable flow* menunjukkan nilai 0 mL/min. Pada bagian water level, nilai setpoint ditetapkan sebesar 6 cm, sedangkan PV pada HMI menunjukkan jarak sensor terhadap permukaan air sebesar 33 cm. Dengan jarak referensi sensor terhadap dasar tangki sebesar 40 cm, tinggi air akhir dapat dihitung sebesar 7 cm. Hasil tersebut menunjukkan bahwa tinggi air akhir berada sekitar 1 cm di atas *setpoint* level. Selisih tersebut dapat terjadi karena masih terdapat air yang mengalir menuju tangki sesaat setelah kondisi *interlock* tercapai

4.6.3. Pengujian *Closed-Loop Setpoint* 2000 mL/min

Pengujian *closed-loop* kedua dilakukan pada *setpoint* 2000 mL/min menggunakan parameter akhir $K_p = 3,5$, $K_i = 3,5$, dan $K_d = 0,35$. Grafik hasil pengujian ditunjukkan pada Gambar 4.31.



Gambar 4.39 Respons *Closed-Loop* pada Setpoint 2000 mL/min

Berdasarkan data pengujian, respons pada SP 2000 mL/min memiliki karakteristik yang berbeda secara signifikan dibandingkan SP 1000 mL/min. Pada fase awal, *process variable* mengalami kenaikan sangat cepat hingga mencapai nilai maksimum 3499,26 mL/min. Berdasarkan pengolahan data sesuai Subbab 3.6.5 diperoleh *rise time* sebesar 1,012 s dan *maximum overshoot* sebesar 74,96%.

Besarnya overshoot pada pengujian ini dapat dijelaskan secara langsung dari data keluaran pengendali. Pada sekitar detik ke-5, ketika *process variable* masih bernilai nol dan error terhadap *setpoint* masih sangat besar, keluaran PWM mencapai 65.535, yaitu batas atas keluaran CTRL_PID. Pada detik berikutnya *process variable* meningkat menjadi sekitar 1690,76 mL/min, kemudian menjadi 3344,58 mL/min, 3486,65 mL/min, dan akhirnya mencapai sekitar 3499,26 mL/min. Sementara *process variable* masih meningkat, keluaran pengendali sebenarnya telah mulai dikurangi dari sekitar 59.748 menjadi 52.208, kemudian

45.756 dan 39.446. Pola tersebut menunjukkan adanya keterlambatan antara perubahan sinyal kendali dan perubahan laju aliran yang terukur.

Fenomena tersebut umum dijumpai pada pengendali PID yang bekerja dengan aktuator yang mempunyai batas keluaran. Ketika error awal besar, pengendali dapat meminta aksi kendali lebih besar daripada yang secara fisik dapat diberikan aktuator sehingga sinyal mencapai kondisi saturasi. Selama plant masih memiliki dinamika dan keterlambatan, *process variable* dapat terus meningkat meskipun keluaran pengendali telah mulai diturunkan. Literatur mengenai PID dengan *actuator saturation* menunjukkan bahwa saturasi dapat meningkatkan *overshoot* dan memperpanjang respons transien.

Namun, pada sistem ini overshoot tersebut tidak tepat jika sepenuhnya disebut sebagai akibat tidak adanya anti-windup. Implementasi CTRL_PID dari OSCAT menggunakan blok internal FT_PIDWL yang didokumentasikan memiliki dynamic wind-up reset dan pembatas keluaran. Artinya, struktur blok memang telah memiliki mekanisme untuk mengurangi efek *wind-up*. Meskipun demikian, mekanisme tersebut tidak dapat menghilangkan fakta bahwa pada awal pengujian pompa tetap menerima perintah maksimum sementara respons fisik dan sinyal sensor memiliki keterlambatan.

Hal yang sangat penting pada SP 2000 mL/min adalah bahwa nilai puncak 3499,26 mL/min hampir tepat berada pada batas skala 3500 mL/min yang digunakan pada pengiriman dan pengolahan data *flow*. Dengan demikian, nilai overshoot 74,96% merupakan *overshoot* berdasarkan skala *process variable* OpenPLC, bukan bukti bahwa laju aliran fisik pompa benar-benar mencapai 3499,26 mL/min. Nilai tersebut juga berpotensi telah mengalami *clipping* karena sistem membatasi representasi *flow* sampai sekitar 3500 mL/min. Oleh karena itu, pembahasan *overshoot* pada titik ini harus dipisahkan dari hasil kalibrasi volumetrik sensor.

Setelah fase transien awal selesai, performa sistem berubah menjadi jauh lebih baik. Pada rentang 30–90 s diperoleh rata-rata *process variable* sebesar 2003,17 mL/min, dengan bias hanya 0,158% terhadap setpoint. Nilai MAE sebesar 38,996 mL/min atau 1,950%, sedangkan RMSE sebesar 55,435 mL/min atau

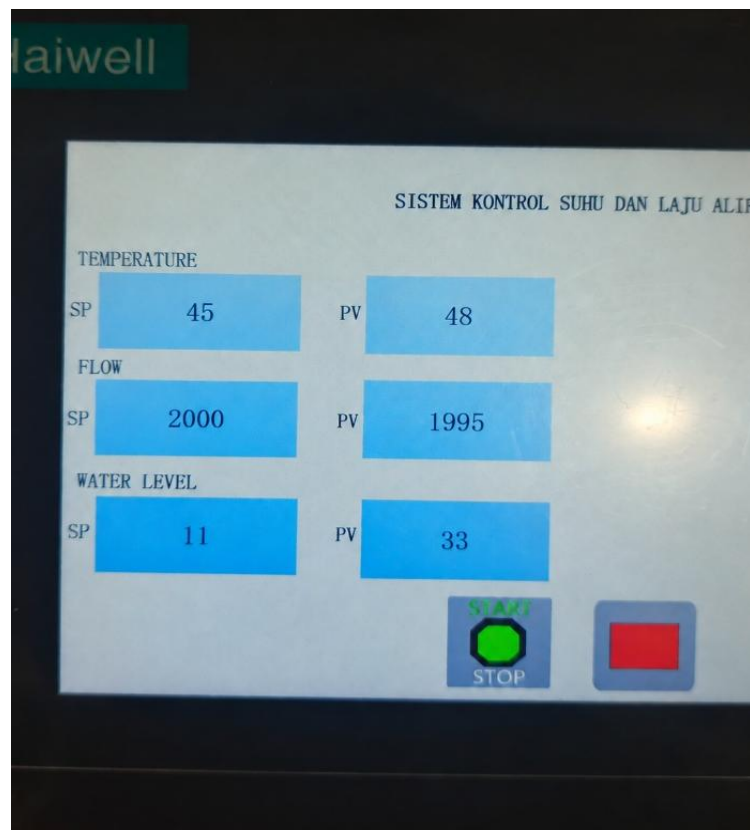
2,772%. Sebanyak 90,20% data berada di dalam batas $\pm 5\%$ dan seluruh data steady-state berada di dalam batas $\pm 10\%$.

Nilai PWM rata-rata pada daerah steady-state berada di sekitar 32.655, jauh di bawah batas maksimum 65.535. Hal ini menunjukkan bahwa saturasi terjadi terutama pada fase awal dan tidak berlangsung secara permanen. Setelah respons awal turun, CTRL_PID mampu mencari daerah operasi yang sesuai dan mempertahankan process variable di sekitar 2000 mL/min.

Ketidakhulusan kecil yang masih terlihat di sekitar setpoint merupakan kombinasi antara perubahan *flow* fisik dan variasi pengukuran. Pada sekitar 2 L/min, YF-S201 secara nominal menghasilkan sekitar 15 pulsa selama interval satu detik. Perubahan satu pulsa masih ekuivalen dengan sekitar 133 mL/min, walaupun secara persentase pengaruhnya lebih kecil daripada pada SP 1000 mL/min. Setelah melalui low-pass filter, fluktuasi tersebut menjadi lebih kecil tetapi tidak sepenuhnya hilang. Penelitian sensor YF-S201 juga menunjukkan bahwa kalibrasi terhadap sistem aktual diperlukan untuk memperoleh pembacaan yang lebih representatif.

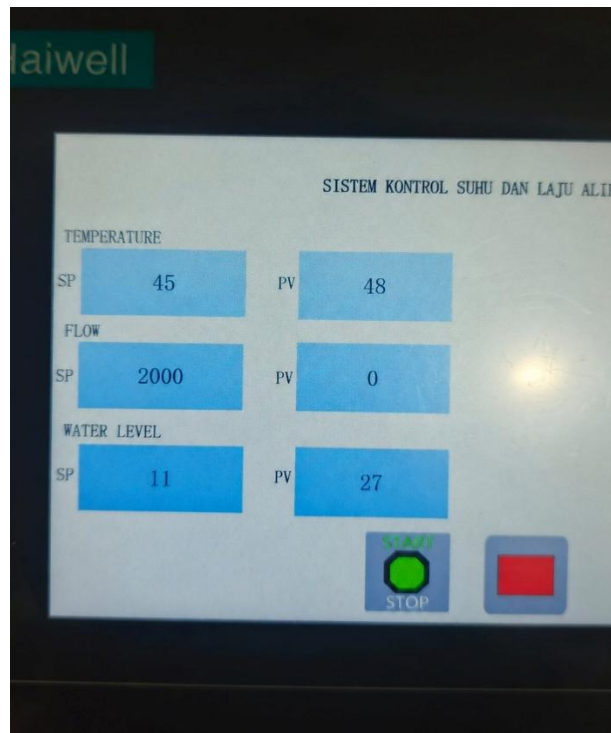
Dengan demikian, overshoot 74,96% pada SP 2000 mL/min terutama merupakan fenomena transien akibat perintah kendali awal yang mencapai batas maksimum, adanya keterlambatan plant dan jalur pengukuran, serta batas skala PV 3500 mL/min. Kondisi tersebut tidak mencerminkan buruknya kinerja steady-state, karena setelah transien berakhir SP 2000 justru menghasilkan performa steady-state terbaik dari ketiga kondisi pengujian.

Dengan demikian, SP 2000 mL/min menunjukkan performa steady-state relatif terbaik dari ketiga kondisi pengujian berdasarkan MAE relatif, RMSE relatif, koefisien variasi, serta persentase data dalam pita toleransi. Namun, kondisi tersebut tidak menunjukkan performa transien terbaik karena *maximum overshoot* tercatat sebesar 74,96% merupakan nilai tertinggi dari ketiga *setpoint*. Oleh karena itu, evaluasi performa perlu membedakan karakteristik transien dengan karakteristik *steady-state*.



Gambar 4.40 Tampilan HMI pada Pengujian *Closed-Loop Setpoint 2000 mL/min*

Gambar 4.32 menunjukkan kondisi HMI pada saat pengujian *closed-loop setpoint flow* 2000 mL/min masih berlangsung. Nilai *process variable flow* yang tampil sebesar 1995 mL/min, sehingga nilai aktual telah sangat dekat dengan setpoint yang diberikan. Pada bagian *water level*, *setpoint* ditetapkan sebesar 11 cm, sedangkan nilai PV yang ditampilkan sebesar 33 cm. Nilai PV tersebut merupakan jarak sensor JSN-SR04T terhadap permukaan air pada saat pengisian berlangsung. Semakin tinggi permukaan air di dalam tangki, semakin kecil jarak yang terbaca oleh sensor. Tampilan tersebut digunakan untuk memperlihatkan kondisi aktual sistem selama proses pengendalian berlangsung.

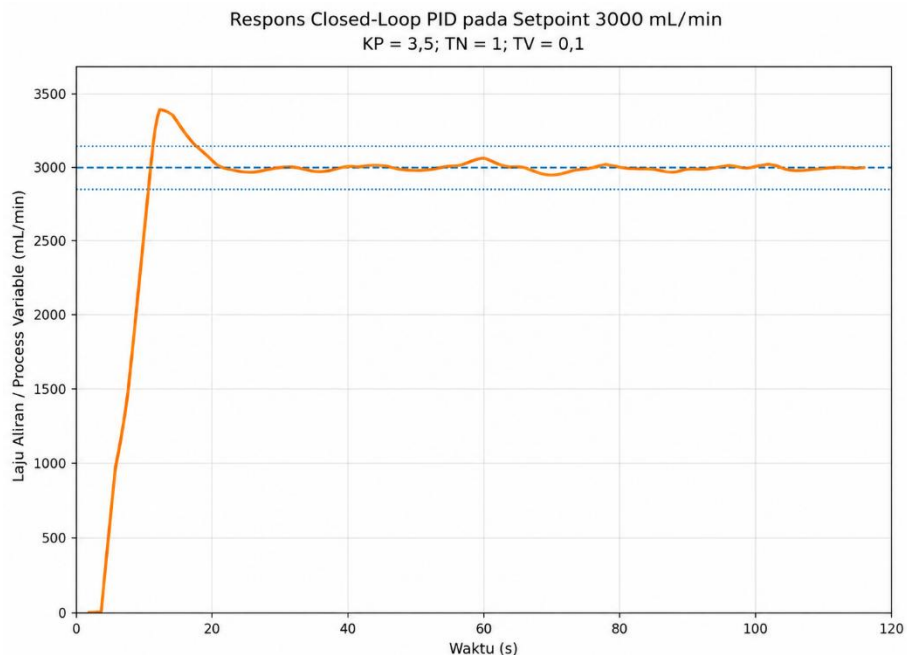


Gambar 4.41 Tampilan HMI Setelah Proses Pengisian pada *Setpoint* 2000 mL/min

Gambar 4.33 menunjukkan kondisi sistem setelah pengisian pada *setpoint* flow 2000 mL/min selesai. Pada kondisi tersebut, nilai *process variable flow* telah menjadi 0 mL/min yang menunjukkan bahwa pompa sudah berhenti. Setpoint level diberikan sebesar 11 cm, sedangkan PV water level pada HMI menunjukkan jarak sensor terhadap permukaan air sebesar 27 cm. Berdasarkan jarak referensi sensor sebesar 40 cm, diperoleh tinggi air akhir sebesar 13 cm. Dengan demikian, tinggi air akhir berada sekitar 2 cm di atas setpoint level. Selisih ini menunjukkan adanya penambahan level sesaat setelah kondisi penghentian tercapai yang dapat dipengaruhi oleh sisa aliran air pada jalur perpipaan dan respons penghentian aktuator

4.6.4. Pengujian *Closed-Loop Setpoint* 3000 mL/min

Pengujian *closed-loop* ketiga dilakukan pada *setpoint* 3000 mL/min menggunakan parameter yang sama, yaitu $K_p = 3,5$, $K_i = 3,5$, dan $K_d = 0,35$. Pengujian ini digunakan untuk mengevaluasi respons sistem pada daerah laju aliran tinggi dan mendekati batas atas plant.



Gambar 4.42 Respons *Closed-Loop* pada Setpoint 3000 mL/min

Berdasarkan hasil pengujian, *process variable* meningkat dengan cepat dari kondisi awal dan mencapai nilai maksimum sekitar 3499,99 mL/min. Berdasarkan pengolahan data menggunakan metode pada Subbab 3.6.5 diperoleh *rise time* sebesar 0,772 s dan *maximum overshoot* sebesar 16,67%.

Respons awal SP 3000 mL/min kembali menunjukkan kondisi saturasi. Pada beberapa sampel awal, keluaran CTRL_PID mencapai batas maksimum 65.535. Pada saat yang sama, variabel analog yang digunakan di dalam OpenPLC juga berada mendekati batas atas representasi internalnya. Kondisi tersebut menunjukkan bahwa jalur pengendalian dan pengukuran berada pada daerah saturasi atau mendekati batas skala yang digunakan dalam progra. Process variable kemudian meningkat menjadi sekitar 3128 mL/min, 3474 mL/min, 3498 mL/min, dan akhirnya berada sangat dekat dengan 3500 mL/min sebelum turun menuju setpoint.

Nilai overshoot 16,67% mempunyai hubungan yang sangat kuat dengan batas maksimum skala pengukuran. Setpoint yang diberikan adalah 3000 mL/min, sedangkan *process variable* dibatasi pada sekitar 3500 mL/min. Oleh sebab itu,

selisih maksimum yang masih dapat ditampilkan hanya sekitar 500 mL/min atau kurang lebih 16,67% terhadap *setpoint*. Artinya, nilai *overshoot* tersebut tidak sepenuhnya menggambarkan puncak alami plant, melainkan sudah dipengaruhi oleh batas maksimum representasi *process variable*. Dengan kata lain, apabila respons sebenarnya melewati daerah tersebut, sistem pencatatan tidak dapat menunjukkan nilai lebih tinggi dari batas yang telah ditetapkan.

Setelah fase awal, *process variable* bergerak di sekitar *setpoint*. Pada rentang 30–90 s diperoleh rata-rata sebesar 3003,32 mL/min, dengan bias hanya sekitar 0,111%. Nilai MAE sebesar 90,86 mL/min atau 3,029% dan RMSE sebesar 117,46 mL/min atau 3,915%. Sebanyak sekitar 78% data berada dalam batas $\pm 5\%$ dan sekitar 98% berada dalam batas $\pm 10\%$.

Walaupun rata-rata *process variable* sangat dekat dengan *setpoint*, data sesaat menunjukkan variasi yang lebih besar daripada SP 2000 mL/min. Pada daerah *steady-state*, *process variable* bergerak sekitar 2776,67–3387,71 mL/min. Hal ini menjelaskan mengapa nilai rata-ratanya dapat sangat baik tetapi settling time dengan batas $\pm 5\%$ tetap cukup panjang, yaitu sekitar 100 s. Dengan toleransi $\pm 10\%$, kondisi stabil diperoleh lebih cepat, sekitar 64 s.

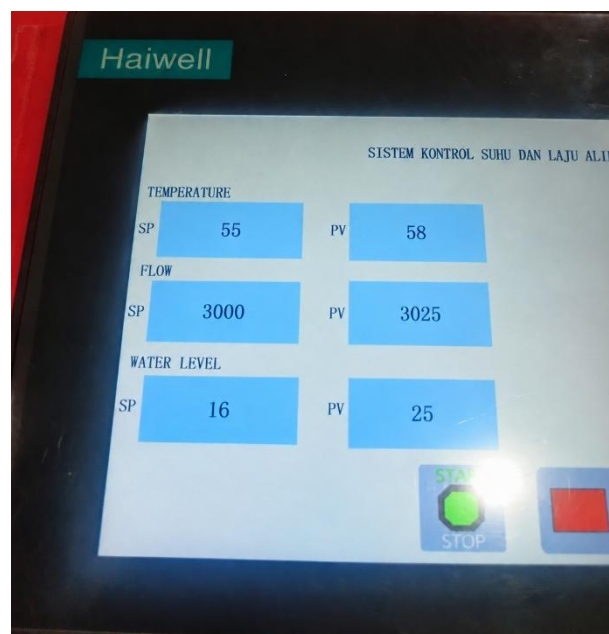
Salah satu penyebabnya adalah sistem bekerja semakin dekat dengan daerah operasi maksimum. Nilai PWM rata-rata pada kondisi *steady-state* berada di sekitar 47.136, lebih tinggi daripada pada SP 1000 dan SP 2000 mL/min. Dengan keluaran dasar yang lebih tinggi, ruang kendali untuk menambah kecepatan pompa menjadi semakin terbatas ketika terjadi penurunan *flow*. Pada sistem pompa berkecepatan variabel, titik kerja tidak hanya ditentukan oleh kecepatan pompa tetapi juga oleh karakteristik hidraulik sistem seperti *head*, rugi-rugi pipa, dan kondisi aliran. Penelitian pada variable-speed pump menunjukkan bahwa perubahan kecepatan dan kondisi sistem menentukan titik operasi aktual pompa

Pada prototipe penelitian ini, kondisi hidraulik juga tidak sepenuhnya konstan selama pengujian. Air berpindah dari satu tangki menuju tangki lainnya sehingga posisi permukaan air berubah terhadap waktu. Perubahan tersebut dapat mengubah beda tekanan atau *static head* yang harus dilawan pompa. Oleh sebab itu, nilai *flow* yang dihasilkan untuk suatu PWM tidak harus identik sepanjang

pengujian, terutama ketika sistem beroperasi pada daerah flow tinggi. Pengaruh ini terjadi bersamaan dengan variasi pembacaan sensor dan aksi koreksi CTRL_PID.

Dari sisi sensor, pada sekitar 3 L/min frekuensi nominal YF-S201 sekitar 22–23 pulsa per detik. Pengaruh satu pulsa terhadap nilai relatif memang lebih kecil daripada pada SP 1000, tetapi secara absolut satu pulsa tetap ekuivalen dengan sekitar 133 mL/min untuk interval penghitungan satu detik. Hal tersebut masih cukup besar untuk terlihat sebagai variasi pada grafik *process variable*, khususnya ketika digabungkan dengan perubahan PWM *closed-loop*.

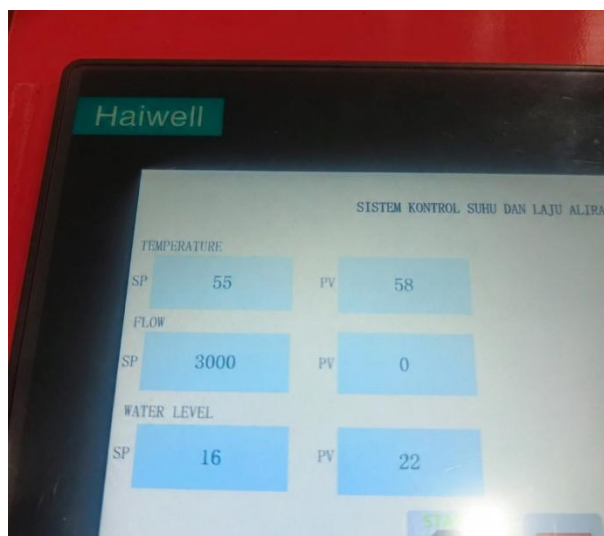
Dengan demikian, respons SP 3000 mL/min menunjukkan bahwa pengendali masih dapat mempertahankan nilai rata-rata *process variable* sangat dekat dengan *setpoint*, tetapi kestabilan sesaat menurun karena sistem bekerja dekat batas atas skala pengukuran dan daerah operasi plant.



Gambar 4.43 Tampilan HMI pada Pengujian *Closed-Loop Setpoint* 3000 mL/min

Gambar 4.35 menunjukkan kondisi HMI pada saat pengujian *closed-loop* dengan *setpoint flow* 3000 mL/min masih berlangsung. Pada kondisi tersebut, nilai *process variable flow* yang ditampilkan pada HMI sebesar 3025 mL/min, sehingga PV flow telah berada di sekitar *setpoint* yang diberikan. Pada bagian *water level*, *setpoint* level ditetapkan sebesar 16 cm, sedangkan nilai PV yang ditampilkan pada

HMI sebesar 25 cm. Nilai PV tersebut merupakan jarak antara sensor JSN-SR04T dan permukaan air, bukan tinggi air secara langsung. Dengan jarak referensi sensor terhadap dasar tangki sebesar 40 cm, tinggi air pada saat pengambilan gambar adalah sekitar 15 cm. Karena nilai tersebut masih berada di bawah LEVEL_SP sebesar 16 cm, kondisi interlock belum tercapai sehingga pompa masih bekerja dan proses pengendalian laju aliran masih berlangsung.



Gambar 4.44 Tampilan HMI Setelah Proses Pengisian pada Setpoint 3000 mL/min

Gambar 4.36 menunjukkan kondisi HMI setelah proses pengisian pada *setpoint flow* 3000 mL/min selesai. Setelah pompa dihentikan, process variable flow terbaca sebesar 0 mL/min. *Setpoint level* yang digunakan sebesar 16 cm, sedangkan PV pada bagian water level menunjukkan jarak sensor terhadap permukaan air sebesar 22 cm. Dengan referensi posisi sensor sebesar 40 cm, diperoleh tinggi air akhir sebesar 18 cm. Nilai tersebut berada sekitar 2 cm di atas setpoint level. Pada kondisi laju aliran yang lebih tinggi, air yang masih berada di dalam jalur perpipaan dapat tetap masuk ke tangki sesaat setelah perintah penghentian diberikan sehingga menyebabkan level akhir sedikit melebihi setpoint

4.6.5. Perbandingan Hasil Pengujian *Closed-Loop*

Setelah dilakukan pengujian *closed-loop* pada *setpoint* 1000 mL/min, 2000 mL/min, dan 3000 mL/min, selanjutnya dilakukan perbandingan untuk mengetahui karakteristik pengendali pada daerah operasi rendah, menengah, dan tinggi. Ketiga

pengujian menggunakan parameter pengendali yang sama, yaitu $K_P = 3,5$, $T_N = 1$, dan $T_V = 0,1$.

Perbandingan dilakukan berdasarkan nilai *rise time*, *maximum overshoot*, rata-rata *process variable* pada kondisi *steady-state*, bias terhadap setpoint, *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), koefisien variasi, serta persentase data yang berada di dalam daerah toleransi. Analisis kondisi *steady-state* dilakukan menggunakan data pada rentang waktu 30–90 detik.

Grafik pada Subbab 4.6.2 sampai 4.6.4 ditampilkan menggunakan moving average tujuh titik untuk memperjelas kecenderungan respons. Proses penghalusan hanya digunakan untuk visualisasi; seluruh parameter numerik pada Tabel 4.8 dihitung menggunakan data pengukuran asli sebelum smoothing.

Tabel 4.8 Perbandingan Hasil Pengujian *Closed-Loop*

Parameter	SP 1000	SP 2000	SP 3000
Rise time (s)	2,051	1,012	0,772
Maximum overshoot (%)	9,36	74,96	16,67
Rata-rata PV (mL/min)	1001,11	2003,17	3003,32
Bias absolut terhadap setpoint (%)	0,111	0,158	0,111
Standar deviasi (mL/min)	49,22	55,90	118,61
Koefisien variasi (%)	4,917	2,790	3,949
MAE (mL/min)	36,17	39,00	90,86

MAE terhadap SP (%)	3,617	1,950	3,029
RMSE (mL/min)	48,75	55,44	117,46
RMSE terhadap SP (%)	4,875	2,772	3,915
Data dalam batas $\pm 5\%$ (%)	70,59	90,20	78,00
Data dalam batas $\pm 10\%$ (%)	96,08	100	98,00

Berdasarkan Tabel 4.8, ketiga kondisi pengujian menunjukkan bahwa rata-rata process variable dapat dipertahankan sangat dekat dengan nilai *setpoint*. Pada SP 1000 mL/min diperoleh rata-rata PV sebesar 1001,11 mL/min, pada SP 2000 mL/min sebesar 2003,17 mL/min, dan pada SP 3000 mL/min sebesar 3003,32 mL/min. Ketiga kondisi memiliki bias rata-rata kurang dari 0,2%. Hasil tersebut menunjukkan bahwa secara rata-rata *Function Block CTRL_PID* mampu melakukan tracking terhadap nilai referensi dengan baik.

Meskipun demikian, kedekatan nilai rata-rata terhadap setpoint tidak dapat digunakan sebagai satu-satunya indikator performa sistem. Nilai rata-rata dapat terlihat sangat baik walaupun process variable masih mengalami variasi di atas dan di bawah setpoint. Oleh karena itu, parameter MAE, RMSE, standar deviasi, serta persentase data yang berada di dalam batas toleransi perlu dianalisis bersama.

Pada setpoint 1000 mL/min diperoleh overshoot sebesar 9,36%. Nilai tersebut merupakan overshoot paling kecil dari ketiga *setpoint*. Namun, pada kondisi steady-state diperoleh koefisien variasi sebesar 4,917%, yang merupakan nilai paling tinggi secara relatif. Hal tersebut menunjukkan bahwa meskipun tidak terjadi lonjakan transien yang besar, *process variable* masih mengalami variasi relatif yang cukup terlihat di sekitar *setpoint*.

Salah satu penyebab kondisi tersebut berkaitan dengan prinsip pembacaan sensor YF-S201 yang menggunakan pulsa. Pada *firmware* digunakan hubungan nominal antara frekuensi pulsa dan laju aliran. Untuk laju aliran sekitar 1 L/min, frekuensi pulsa yang dihasilkan secara nominal hanya sekitar 7–8 pulsa per detik. Dengan interval pengukuran sekitar satu detik, perubahan satu pulsa dapat menghasilkan perubahan hasil perhitungan yang relatif besar. Oleh karena itu, pengaruh diskritisasi pulsa menjadi lebih dominan pada daerah laju aliran rendah.

Karakteristik tersebut sejalan dengan penelitian Sitompul et al. yang menjelaskan bahwa YF-S201 menghasilkan pulsa berdasarkan putaran rotor dan hubungan antara jumlah pulsa dengan laju aliran perlu dikalibrasi karena tidak sepenuhnya linear pada seluruh rentang pengoperasian.

Pada setpoint 2000 mL/min diperoleh *overshoot* paling besar, yaitu 74,96%. Nilai tersebut terjadi pada fase awal pengujian ketika keluaran pengendali mencapai daerah maksimum dan *process variable* meningkat hingga mendekati batas pembacaan 3500 mL/min. *Overshoot* yang besar pada bagian awal tidak menunjukkan bahwa kondisi steady-state buruk. Setelah fase transien berakhir, *process variable* justru menunjukkan performa *steady-state* paling baik dibandingkan dua setpoint lainnya.

Hal tersebut terlihat dari nilai MAE relatif sebesar 1,950% dan RMSE relatif sebesar 2,772%, yang merupakan nilai paling kecil dari ketiga pengujian. Selain itu, sebanyak 90,20% data steady-state berada di dalam batas $\pm 5\%$ dan seluruh data berada di dalam batas $\pm 10\%$. Koefisien variasi sebesar 2,790% juga merupakan nilai paling rendah.

Berdasarkan parameter *steady-state*, *setpoint* 2000 mL/min memberikan performa relatif terbaik dari ketiga kondisi pengujian. Hal tersebut ditunjukkan oleh MAE relatif, RMSE relatif, koefisien variasi, dan persentase data dalam pita toleransi yang lebih baik dibandingkan setpoint lainnya. Namun, dari sisi respons transien, *setpoint* 2000 mL/min menghasilkan *maximum overshoot* tercatat paling besar. Oleh karena itu, performa steady-state yang baik tidak secara langsung menunjukkan bahwa karakteristik transiennya juga merupakan yang terbaik. Pada daerah ini jumlah pulsa sensor sudah lebih banyak dibandingkan kondisi 1000

mL/min sehingga pengaruh perubahan satu pulsa menjadi lebih kecil secara persentase. Di sisi lain, pompa juga belum bekerja sedekat kondisi maksimum seperti pada setpoint 3000 mL/min.

Overshoot awal yang besar pada SP 2000 mL/min juga berkaitan dengan kondisi saturasi keluaran. Pada sistem kontrol dengan aktuator yang memiliki batas maksimum, perubahan setpoint yang besar dapat menyebabkan keluaran pengendali mencapai batas aktuator. Kondisi saturasi dapat meningkatkan overshoot dan memperpanjang karakteristik transien, terutama ketika plant masih memberikan respons setelah keluaran pengendali mulai diturunkan. Fenomena saturasi pada pengendali PID telah dibahas oleh Bohn dan Atherton sebagai salah satu faktor penting dalam respons sistem kontrol praktis

Pada setpoint 3000 mL/min diperoleh overshoot sebesar 16,67%. Nilai ini lebih kecil dibandingkan SP 2000 mL/min, tetapi kondisi operasi berada lebih dekat dengan batas atas plant dan sistem pengukuran. Nilai process variable maksimum mendekati 3500 mL/min, yaitu batas skala *flow* yang digunakan pada program. Oleh sebab itu, nilai puncak yang tercatat perlu diinterpretasikan dengan hati-hati karena telah dipengaruhi oleh batas representasi sinyal.

Pada kondisi *steady-state*, rata-rata *process variable* sebesar 3003,32 mL/min menunjukkan bahwa nilai rata-rata masih sangat dekat terhadap setpoint. Namun, standar deviasi sebesar 118,61 mL/min dan RMSE sebesar 117,46 mL/min menunjukkan bahwa perubahan sesaat lebih besar dibandingkan dua pengujian lainnya.

Koefisien variasi sebesar 3,949% masih lebih rendah dibandingkan SP 1000 mL/min, tetapi lebih besar dibandingkan SP 2000 mL/min. Hal ini menunjukkan bahwa pada daerah operasi tinggi, faktor keterbatasan plant mulai memberikan pengaruh yang lebih besar.

Dari sisi fisik, peningkatan kecepatan pompa tidak selalu menghasilkan peningkatan flow secara linear. Titik kerja pompa ditentukan oleh interaksi antara karakteristik pompa dan karakteristik sistem perpipaan, termasuk beda tekanan, rugi-rugi pipa, fitting, serta perubahan kondisi hidraulik. Penelitian pada pompa dengan pengaturan kecepatan menunjukkan bahwa perubahan kecepatan pompa

mengubah titik kerja, tetapi nilai *flow* aktual tetap dipengaruhi oleh karakteristik sistem tempat pompa beroperasi.

Jika ketiga setpoint dibandingkan berdasarkan rise time, terlihat bahwa rise time semakin kecil ketika *setpoint* dinaikkan, yaitu 2,051 s pada SP 1000 mL/min, 1,012 s pada SP 2000 mL/min, dan 0,772 s pada SP 3000 mL/min. Hal tersebut terjadi karena pada error awal yang lebih besar, CTRL_PID memberikan aksi kendali yang lebih besar dan pada SP 2000 serta SP 3000 keluaran pengendali dengan cepat menuju daerah maksimum. Akibatnya, pompa memperoleh perintah awal yang lebih besar sehingga *process variable* lebih cepat melewati batas 90% setpoint.

Namun, *rise time* yang semakin cepat tidak selalu menunjukkan performa yang semakin baik. Respons yang terlalu cepat dapat diikuti overshoot yang lebih besar apabila energi atau aksi kendali yang diberikan pada fase awal terlalu tinggi. Hal tersebut terlihat jelas pada SP 2000 mL/min, yang memiliki rise time hanya 1,012 s tetapi menghasilkan *overshoot* 74,96%.

Berdasarkan keseluruhan parameter, pengujian SP 2000 mL/min memberikan performa steady-state relatif terbaik, sedangkan SP 1000 mL/min memberikan *overshoot* paling rendah. Pada SP 3000 mL/min, pengendali masih mampu mempertahankan rata-rata *process variable* mendekati setpoint, tetapi respons mulai dipengaruhi oleh keterbatasan operasi plant dan batas atas sistem pengukuran.

Secara keseluruhan, parameter:

$$KP = 3,5, TN = 1, TV = 0,1$$

Mampu digunakan pada ketiga titik kerja dengan karakteristik performa yang berbeda. Hasil ini menunjukkan bahwa satu set parameter PID dapat memberikan tracking rata-rata yang baik pada rentang pengujian, tetapi respons transien dan tingkat fluktuasi tetap bergantung pada titik operasi plant.

Perlu ditegaskan bahwa hasil pada Subbab 4.6 merupakan evaluasi kemampuan *process variable* OpenPLC dalam mengikuti *setpoint*. Kedekatan PV terhadap *setpoint* belum secara langsung menunjukkan bahwa laju aliran fisik

memiliki nilai yang identik. Kesesuaian tersebut dievaluasi secara terpisah melalui pengujian volumetrik sensor *flow*.

4.6.6. Analisis Fluktuasi Hasil Pengujian *Closed-Loop*

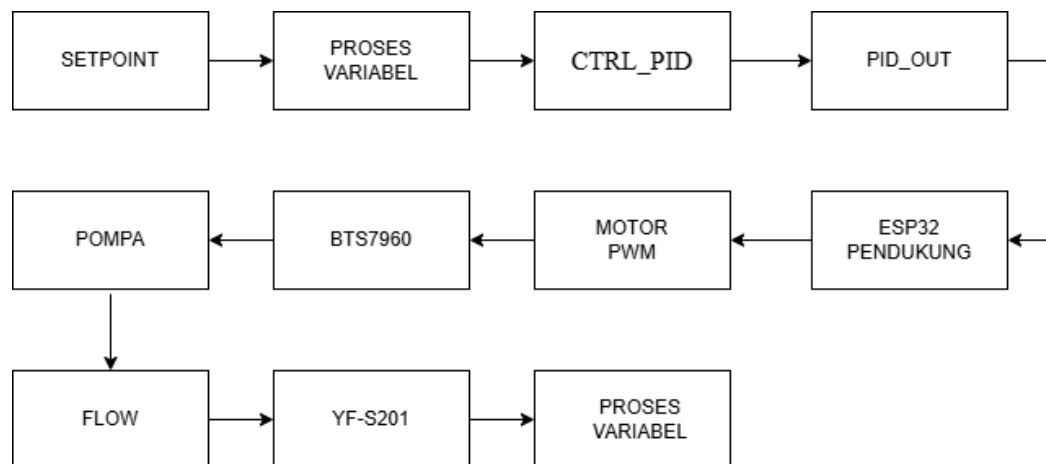
Berdasarkan grafik respons pada *setpoint* 1000 mL/min, 2000 mL/min, dan 3000 mL/min, terlihat bahwa respons *closed-loop* memiliki bentuk yang lebih berfluktuasi dibandingkan respons pada pengujian *open-loop*. Kondisi tersebut tidak secara langsung menunjukkan adanya ketidakstabilan pada OpenPLC maupun kegagalan pengendali PID, tetapi merupakan hasil gabungan dari karakteristik sistem umpan balik, metode pengukuran sensor, proses konversi sinyal, dinamika pompa, serta kondisi hidraulik plant.

Pada sistem *open-loop*, nilai PWM yang diberikan kepada pompa dipertahankan tetap selama pengujian. Setelah fase transien selesai, plant menerima masukan yang relatif konstan. Dengan kondisi tersebut, kecepatan pompa juga cenderung tetap sehingga nilai laju aliran bergerak menuju satu daerah kondisi tunak. Hal ini menyebabkan grafik *open-loop* terlihat lebih halus setelah respons awal selesai.

Kondisi berbeda terjadi pada sistem *closed-loop*. Pada sistem ini, PWM tidak bernilai tetap, tetapi terus diperbarui berdasarkan perbedaan antara *setpoint* dan *process variable*. Ketika *process variable* lebih kecil daripada *setpoint*, pengendali akan meningkatkan aksi kendali. Sebaliknya, ketika *process variable* lebih tinggi daripada *setpoint*, pengendali akan menurunkan keluaran.

Dengan demikian, perubahan kecil pada *process variable* dapat memicu perubahan PWM. Perubahan PWM tersebut kembali mengubah kecepatan pompa dan selanjutnya mengubah laju aliran yang dibaca oleh sensor. Proses tersebut berlangsung secara berulang selama sistem beroperasi.

Secara umum proses koreksi tersebut dapat digambarkan sebagai:



Gambar 4.45 Diagram Blok Proses Koreksi Error

Mekanisme umpan balik tersebut menyebabkan sinyal closed-loop secara alami memiliki aktivitas yang lebih tinggi daripada *open-loop*. Oleh karena itu, grafik *closed-loop* tidak diharapkan memiliki bentuk sehalus grafik open-loop dengan PWM konstan.

Selain aksi pengendali, sensor YF-S201 juga memberikan kontribusi terhadap variasi *process variable*. Sensor ini bekerja dengan rotor dan *Hall effect*, sehingga aliran air menghasilkan pulsa digital. Pulsa tersebut kemudian dihitung oleh ESP32 dalam periode tertentu dan dikonversi menjadi nilai laju aliran. Sekitar 0,133 L/min atau 133 mL/min untuk interval penghitungan sekitar satu detik.

Nilai tersebut cukup besar apabila dibandingkan dengan setpoint 1000 mL/min. Pada *flow* sekitar 1000 mL/min, jumlah pulsa nominal hanya sekitar 7–8 pulsa dalam satu detik. Dengan demikian, perubahan satu pulsa dapat memberikan perubahan sekitar 13% terhadap nilai *flow*.

Pada *flow* 2000 mL/min, jumlah pulsa nominal meningkat menjadi sekitar 15 pulsa per detik sehingga pengaruh satu pulsa secara relatif berkurang menjadi sekitar 6–7%. Pada *flow* 3000 mL/min, jumlah pulsa nominal sekitar 22–23 pulsa per detik sehingga efek kuantisasi pulsa secara relatif menjadi lebih kecil lagi.

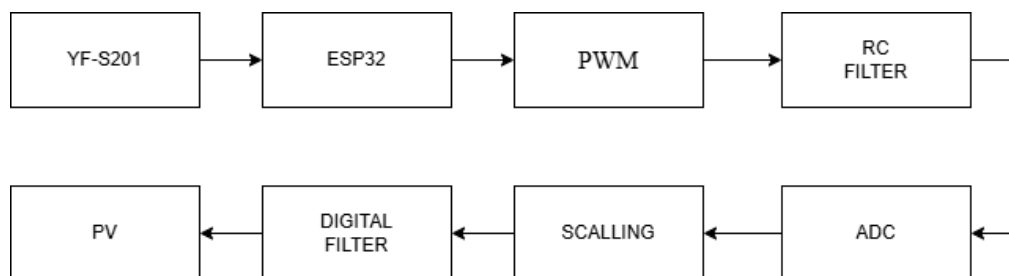
Karakteristik tersebut membantu menjelaskan mengapa variasi relatif pada SP 1000 mL/min masih cukup besar walaupun rata-rata *process variable* berada

sangat dekat dengan setpoint. Sensor YF-S201 tidak menghasilkan nilai flow kontinu, melainkan nilai yang berasal dari pulsa diskrit.

Penelitian Sitompul et al. juga menunjukkan bahwa YF-S201 bekerja dengan menghasilkan pulsa dari putaran rotor dan hubungan pulsa terhadap *flow* tidak sepenuhnya linear, sehingga proses kalibrasi diperlukan untuk mendapatkan hasil pengukuran yang lebih representatif [41]. Penelitian Rashad et al. juga menunjukkan bahwa proses kalibrasi dapat meningkatkan kesesuaian hasil pengukuran YF-S201 terhadap volume referensi [44].

Pada sistem penelitian ini, data dari YF-S201 juga tidak langsung digunakan oleh CTRL_PID. Data terlebih dahulu melalui beberapa tahap pengolahan, yaitu pembacaan pulsa oleh ESP32, konversi nilai *flow*, pemetaan menjadi PWM 12-bit, proses RC low-pass filter, pembacaan ADC pada ESP32 OpenPLC, proses scaling, dan low-pass filter digital.

Secara umum, rantai pengukuran flow dapat dituliskan sebagai:



Gambar 4.46 Diagram Blok Proses Pengukuran Flow

Tahapan tersebut berfungsi agar data dapat diteruskan antarperangkat dan fluktuasi frekuensi tinggi dapat dikurangi. Namun, setiap proses *filtering* juga menghasilkan keterlambatan tertentu terhadap perubahan sinyal.

Pada *low-pass* filter digital yang digunakan, data baru diberikan bobot sebesar 0,2 sedangkan data sebelumnya diberikan bobot sebesar 0,8. Dengan struktur tersebut, perubahan flow tidak langsung diteruskan sepenuhnya ke process variable, tetapi bergerak secara bertahap menuju nilai baru.

Kondisi tersebut membantu menghasilkan pembacaan yang lebih stabil, tetapi pada saat yang sama menambah *lag*. Ketika PID mengubah PWM, respons fisik plant memerlukan waktu untuk berubah, kemudian sensor membutuhkan

waktu untuk membaca perubahan tersebut, dan filter kembali memberikan keterlambatan sebelum nilai baru digunakan oleh CTRL_PID.

Literatur mengenai PID menjelaskan bahwa *noise* pengukuran dapat menyebabkan aktivitas kontrol yang tidak diinginkan. *Filtering* dapat mengurangi efek tersebut, tetapi *filtering* yang terlalu kuat dapat menurunkan performa akibat bertambahnya keterlambatan. Oleh sebab itu terdapat kompromi antara pengurangan *noise* dan kecepatan respons kontrol [39].

Berdasarkan kondisi tersebut, solusi untuk membuat grafik lebih halus tidak dapat dilakukan hanya dengan memperbesar konstanta filter. Apabila filter dibuat terlalu kuat, grafik memang dapat terlihat lebih tenang, tetapi *process variable* menjadi lebih lambat mengikuti perubahan *flow* aktual. Kondisi tersebut bahkan dapat menyebabkan PID melakukan koreksi berdasarkan informasi yang sudah tertunda.

Faktor berikutnya adalah saturasi keluaran pengendali. Pada pengujian SP 2000 dan SP 3000 mL/min, keluaran PID pada fase awal mencapai nilai maksimum. Ketika keluaran mencapai batas tersebut, peningkatan error tidak lagi dapat menghasilkan peningkatan aktuasi yang sebanding karena pompa telah menerima perintah maksimum.

Bohn dan Atherton menjelaskan bahwa saturasi aktuator merupakan salah satu nonlinieritas yang umum pada sistem kontrol dan dapat menghasilkan *overshoot* besar serta meningkatkan waktu pemulihan setelah perubahan *setpoint* [40].

Pada SP 2000 mL/min, kondisi tersebut terlihat dari *overshoot* sebesar 74,96%. Meskipun pengendali mulai mengurangi PWM setelah *process variable* meningkat, laju aliran masih terus naik akibat adanya keterlambatan plant dan sistem pengukuran. Akibatnya, nilai PV melewati *setpoint* sebelum akhirnya turun menuju kondisi stabil.

Pada SP 3000 mL/min, respons juga mencapai batas atas *process variable* sekitar 3500 mL/min. Kondisi ini menyebabkan analisis nilai puncak perlu dilakukan secara hati-hati karena skala pembacaan sudah mendekati batas maksimum yang ditetapkan pada program.

Selain aspek elektronik dan kontrol, fluktuasi juga dipengaruhi oleh karakteristik fisik sistem aliran. Pompa tidak bekerja sebagai sumber flow ideal. Nilai laju aliran ditentukan oleh interaksi antara karakteristik pompa dengan hambatan sistem perpipaan.

Hambatan tersebut dapat berasal dari diameter pipa, panjang pipa, fitting, sensor flow, sambungan, beda ketinggian, dan kondisi permukaan air pada tangki. Ketika permukaan air berubah, kondisi tekanan statis yang dihadapi pompa juga dapat berubah.

Pada pompa dengan pengaturan kecepatan, perubahan kecepatan motor akan menggeser titik kerja pompa. Akan tetapi, titik operasi aktual tetap dipengaruhi oleh karakteristik sistem tempat pompa digunakan. Penelitian Dunca et al. menunjukkan bahwa pada pompa dengan kecepatan variabel, hubungan antara kecepatan pompa, head, dan flow menentukan titik kerja aktual sistem [42].

Faktor ini menjadi semakin penting pada SP 3000 mL/min karena pompa bekerja mendekati daerah operasi atas. Ketika ruang untuk meningkatkan kecepatan pompa sudah semakin kecil, perubahan kondisi hidraulik dapat memberikan pengaruh lebih nyata terhadap process variable.

Sementara itu, pada SP 1000 mL/min pengaruh keterbatasan pompa belum dominan, tetapi pengaruh resolusi pulsa sensor menjadi lebih besar. Pada SP 2000 mL/min kedua kondisi tersebut relatif seimbang: jumlah pulsa sensor sudah cukup banyak dan pompa belum bekerja sedekat batas maksimum. Kondisi tersebut sesuai dengan hasil pengujian yang menunjukkan koefisien variasi dan RMSE relatif paling kecil pada SP 2000 mL/min.

Faktor lain yang dapat memengaruhi bentuk grafik adalah interval sampling. Pada sistem digital, data tidak diperoleh secara kontinu tetapi pada waktu-waktu diskrit. Apabila interval pencatatan atau pembacaan tidak selalu identik, posisi titik pada grafik dapat terlihat kurang seragam. Penelitian mengenai *sampling jitter* menunjukkan bahwa variasi interval sampling dapat memengaruhi evaluasi performa loop kontrol digital [43].

Pada penelitian ini, proses penghalusan grafik dilakukan menggunakan metode *moving average* tujuh titik. Metode tersebut digunakan hanya untuk

membantu pembaca melihat kecenderungan utama respons. Data asli tidak dihapus dan tetap digunakan sebagai dasar seluruh perhitungan parameter performa.

Penggunaan grafik yang diperhalus tidak boleh digunakan untuk menghitung ulang peak value, overshoot, rise time, settling time, MAE, maupun RMSE karena penghalusan dapat menurunkan nilai puncak dan mengubah posisi waktu respons. Dengan demikian, grafik hasil *moving average* merupakan representasi visual, sedangkan hasil numerik tetap merepresentasikan data aktual yang direkam oleh sistem.

Berdasarkan keseluruhan analisis, fluktuasi respons closed-loop yang terlihat pada grafik merupakan akibat interaksi beberapa faktor, yaitu aksi koreksi PID yang berlangsung secara kontinu, karakteristik pulsa sensor YF-S201, proses sampling, rantai konversi PWM–RC filter–ADC, low-pass filter digital, saturasi keluaran, dinamika pompa, serta kondisi hidraulik plant.

Berdasarkan keseluruhan analisis, fluktuasi yang terlihat pada respons closed-loop merupakan hasil interaksi beberapa elemen di dalam loop pengendalian, meliputi aksi koreksi PID, karakteristik pulsa YF-S201, proses sampling, konversi PWM–RC–ADC, low-pass filter digital, saturasi keluaran, dinamika pompa, serta kondisi hidraulik plant. Oleh karena itu, bentuk grafik process variable saja tidak cukup untuk mengatribusikan fluktuasi secara khusus kepada OpenPLC. OpenPLC pada sistem ini berfungsi mengeksekusi program pengendalian, sedangkan respons akhir yang terukur merupakan hasil interaksi antara pengendali, sistem pengukuran, aktuator, dan plant.

Secara keseluruhan, hasil pengujian menunjukkan bahwa OpenPLC Runtime pada ESP32 dapat menjalankan Function Block CTRL_PID dan membentuk sistem pengendalian laju aliran air secara closed-loop. Jalur feedback dari YF-S201 menuju OpenPLC dan jalur keluaran pengendali menuju ESP32 pendukung telah mampu meneruskan perubahan sinyal sesuai fungsi masing-masing. Pengujian open-loop juga berhasil memberikan karakteristik plant yang digunakan sebagai dasar penentuan parameter awal PID menggunakan metode Ziegler–Nichols open-loop.

Pada pengujian closed-loop, satu konfigurasi parameter $K_P = 3,5$, $T_N = 1$ s, dan $T_V = 0,1$ s dapat digunakan pada tiga setpoint pengujian, yaitu 1000, 2000, dan 3000 mL/min. Rata-rata process variable pada ketiga kondisi berada dekat dengan setpoint, tetapi karakteristik transien dan fluktuasi steady-state berbeda pada setiap daerah operasi. Hasil tersebut menunjukkan bahwa performa sistem tidak hanya dipengaruhi oleh parameter PID, tetapi juga oleh karakteristik sensor, proses konversi sinyal, aktuator, dan plant. Dengan demikian, evaluasi kemampuan tracking process variable perlu dibedakan dari evaluasi akurasi laju aliran fisik yang memerlukan pengujian volumetrik pada kondisi yang sama.