

TUGAS AKHIR - TF 181801

**RANCANG BANGUN SISTEM PENGENDALIAN LEVEL
MINI PLANT SEPARATOR TIGA FASA MENGGUNAKAN
NEURAL NETWORK PID BERBASIS *INTERNET OF
THINGS***

SULTHAN ARIQ PRANANDA

NRP 02311840000036

Dosen Pembimbing

Dr. Suyanto, S.T., M.T.

NIP 19711113319951002

Program Studi Sarjana

Departemen Teknik Fisika

Fakultas Teknologi Industri dan Rekayasa Sistem

Institut Teknologi Sepuluh Nopember

Surabaya

2022

Halaman ini sengaja dikosongkan



FINAL PROJECT - TF 181801

***NEURAL NETWORK PID CONTROL SYSTEM DESIGN ON THREE
PHASE SEPARATOR MINI PLANT BASED ON INTERNET OF THINGS***

SULTHAN ARIQ PRANANDA

NRP. 0231184000036

Supervisors:

Dr. Suyanto, S.T., M.T.

DEPARTMENT OF ENGINEERING PHYSICS

Faculty of Industrial Technology and System Engineering

Institut Teknologi Sepuluh Nopember

Surabaya 2022

Halaman ini sengaja dikosongkan

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini.

Nama : Sulthan Ariq Prananda
NRP : 0231184000036
Departemen / Prodi : Teknik Fisika / S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "**RANCANG BANGUN SISTEM PENGENDALIAN LEVEL MINI PLANT SEPARATOR TIGA FASA MENGGUNAKAN NEURAL NETWORK PID BERBASIS INTERNET OF THINGS**" adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya-benarnya.

Surabaya, 1 Juli 2022

Yang membuat pernyataan,



Sulthan Ariq Prananda

NRP. 0231184000036

Halaman ini sengaja dikosongkan

**LEMBAR PENGESAHAN
TUGAS AKHIR**

**RANCANG BANGUN SISTEM PENGENDALIAN LEVEL *MINI PLANT*
SEPARATOR TIGA FASA MENGGUNAKAN *NEURAL NETWORK* PID
BERBASIS *INTERNET OF THINGS***

Oleh:

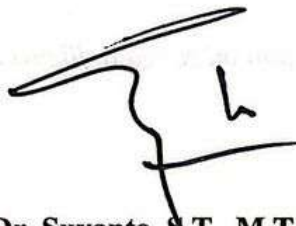
Sulthan Ariq Prananda

NRP. 0231184000036

Surabaya,

Menyetujui,

Pembimbing I



Dr. Suyanto, S.T., M.T.

NIP. 19711113 199512 1 002

Mengetahui,

Kepala Departemen

Teknik Fisika FT- IRS ITS



Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM PENGENDALIAN LEVEL *MINI PLANT*
SEPARATOR TIGA FASA MENGGUNAKAN *NEURAL NETWORK* PID
BERBASIS *INTERNET OF THINGS***

Diajukan Untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Teknik

pada

Program Studi S-1 Departemen Teknik Fisika

Fakultas Teknologi Industri & Rekayasa Sistem (FT-IRS)

Institut Teknologi Sepuluh Nopember

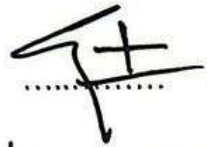
Oleh:

SULTHAN ARIQ PRANANDA

NRP. 0231184000036

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr. Suyanto, S.T., M.T.



(Pembimbing I)

2. Dr. Bambang Lelono Widjiantoro, S.T., M.T.



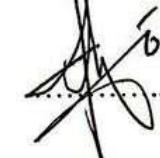
(Ketua Penguji)

3. Moh Kamalul Wafi, S.T., MSc.DIC



(Penguji I)

4. Prof. Dr. Ir. Sekartedjo, M.Sc.



(Penguji II)

SURABAYA

2022

Halaman ini sengaja dikosongkan

RANCANG BANGUN SISTEM PENGENDALIAN LEVEL *MINI PLANT SEPARATOR* TIGA FASA MENGGUNAKAN *NEURAL NETWORK* PID BERBASIS INTERNET OF THINGS

Nama : Sulthan Ariq Prananda
NRP : 0231184000036
Departemen : Teknik Fisika FTIRS - ITS
Dosen Pembimbing : Dr. Suyanto S.T., M.T.

ABSTRAK

Proses dalam industri *oil & gas* terdapat proses separasi minyak mentah pada instrumen separator tiga fasa. Separator perlu dilakukan pengendalian agar produksi minyak stabil dan hasil yang maksimal. Penelitian ini dilakukan perancangan sistem pengendalian PID yang bersifat adaptif menggunakan *Neural Network* yang akan melakukan tuning PID secara adaptif selama proses separasi berlangsung. Sistem kendali separator ini dapat dikendalikan secara jarak jauh karena sistem dirancang dengan kombinasi antara sistem *WSAN* menggunakan modul *Long Range (LoRa)* serta *Internet of Things (IoT)* yang terintegrasi dengan sistem *cloud database*. Sistem dirancang menggunakan protokol IEEE 802.15.4g dan protokol HTTP. Hasil penelitian sistem komunikasi *WSAN* dengan device *LoRa* dapat berkomunikasi secara *real-time* pada jarak 100 m dengan *delay time* 145 ms. Pada pengendalian level campuran respon sistem menghasilkan selisih *rise time* 12.374 detik lebih rendah dan *settling time* lebih rendah 29.445 detik dari pengendali PID. Pada pengendalian level minyak respon sistem menghasilkan selisih *rise time* 16.768 detik lebih rendah dan *settling time* lebih rendah 101.052 detik dari pengendali PID. Kedua pengendali tidak memiliki perbedaan pada *overshoot* maksimum dan *error steady state*.

Kata Kunci: IoT, *Neural Network* PID, Separator Tiga Fasa, *WSAN*

Halaman ini sengaja dikosongkan

***NEURAL NETWORK PID CONTROL SYSTEM DESIGN ON THREE
PHASE SEPARATOR MINI PLANT BASED ON INTERNET OF
THINGS***

Name : Sulthan Ariq Prananda
NRP : 0231184000036
Department : Engineering Physics FTIRS - ITS
Supervisors : Dr. Suyanto S.T., M.T.

ABSTRACT

Process in oil & gas industry has separation process in a three-phase separator instrument. The separator needs to be controlled so that the oil production is stable and get maximum results. In this study, an adaptive PID control system was designed using a Neural Network which will adjust the PID adaptively during separation process. This control system can be controlled remotely because the system is designed with a combination of WSN system using Long Range (LoRa) modules and Internet of Things (IoT) modules that are integrated with a cloud database system. The system is designed using protocol IEEE 802.15.4g and HTTP. The results showed that the WSN communication system with LoRa devices can communicate in real-time at 100 meters with average delay time is 145 ms. When mixed level controlled, the dynamic response system produces a rise time of 12.374 seconds and a settling time of 29.445 seconds lower than the PID controller. When control oil level, the dynamic response system produces a rise time of 16.768 seconds and a settling time of 101.052 seconds lower than the PID controller. Both control systems have no difference in maximum overshoot and steady state error.

Keywords: IoT, Neural Network PID, Three Phase Separator, WSN

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puji selalu disampaikan atas kehadiran Allah SWT Yang Maha Pengasih lagi Maha Penyayang sehingga atas rahmat dan hidayah-Nya laporan tugas akhir yang berjudul **“Rancang Bangun Sistem Pengendalian Level *Mini Plant Separator* Tiga Fasa Menggunakan *Neural Network* PID Berbasis *Internet of Things*”** dapat terselesaikan dengan baik.

Dalam penyusunan laporan tugas akhir ini, banyak pihak-pihak yang turut peran didalamnya. Maka dari itu, secara khusus penulis ingin mengucapkan terima kasih kepada:

1. Bapak Dr. Suyanto, S.T., M.T., selaku Dosen Pembimbing dan Kepala Departemen Teknik Fisika ITS yang telah memberikan ilmu, waktu, bimbingan dan fasilitas selama menempuh pendidikan di Teknik Fisika ITS.
 2. Bapak Dr. Bambang Lelono Widjiantoro, S.T., M.T., Bapak Moh Kamalul Wafi, S.T., MSc.DIC, dan Bapak Prof. Dr. Ir. Sekartedjo, M.Sc., selaku penguji tugas akhir yang telah memberikan masukan kepada penulis tugas akhir dapat terselesaikan dengan baik.
 3. Segenap keluarga, terutama orang tua, serta kekasih penulis yang telah memberikan dukungan serta kasih sayang selama menempuh Pendidikan di ITS.
 4. Ibu Dr. Eng. Nur Laila Hamida, S.T., M.Sc. dan Alm. Bapak Hendra Cordova, S.T., M.T., selaku dosen wali saya yang banyak memberikan saran, bimbingan serta meluangkan waktu untuk keperluan akademik saya selama menempuh pendidikan di Teknik Fisika ITS.
 5. Seluruh dosen, karyawan, dan civitas akademika ITS yang telah memberikan kesempatan, bantuan, dan kerjasamanya selama menempuh Pendidikan di ITS.
 6. Teman-teman Admin ECS, Nogogeni ITS Team, dan Angkatan 2018 Trident Firmusku yang telah memberikan wadah untuk berkembang, memberikan dukungan, kesan, pesan, maupun kenangan yang tak terlupakan selama menempuh masa perkuliahan, terima kasih untuk semuanya.
 7. Teman-teman TAdika mesra, Dwiki dan Andrew, yang telah membantu memberi masukan dan saran serta momen selama pengerjaan tugas akhir.
- Serta pihak-pihak lain yang tidak dapat disebutkan satu-persatu.

Surabaya, 11 Juli 2022

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
<i>COVER PAGE</i>	iii
PERNYATAAN BEBAS PLAGIASI	v
LEMBAR PENGESAHAN	vii
LEMBAR PENGESAHAN	ix
ABSTRAK.....	xi
<i>ABSTRACT</i>	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL.....	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Laporan.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Proses Separasi Minyak Tiga Fasa.....	5
2.2 Internet of Things	6
2.2.1 Arsitektur IoT	6
2.2.2 Elemen IoT	7
2.3 <i>Industrial Wireless Sensor and Actuator Networks</i>	8
2.4 Firebase Real Time Database.....	8
2.5 Sistem Pengendalian PID.....	9
2.5.1 Tuning Parameter PID untuk Level Control	11

2.6	Analisis Respon Dinamik Sistem	14
2.7	Jaringan Syaraf Tiruan (<i>Artificial Neural Network</i>).....	14
2.7.1	Algoritma Pelatihan <i>Backpropagation</i>	17
2.8	Mikrokontroler ESP32	19
2.9	Barometric Pressure Sensor MS5611	19
2.10	Servo Motor Control Valve.....	20
2.11	Transceiver <i>Wireless</i> Long Range RFM95	21
BAB III METODOLOGI PENELITIAN		23
3.1	Studi literatur.....	23
3.2	Desain Mini Plant Separator.....	24
3.3	Perancangan <i>Human Machine Interfaces</i> dan Sistem Komunikasi <i>plant</i>	29
3.3.1	Perancangan Komunikasi <i>Wireless Sensor and Actuator Networks</i>	30
3.3.2	Perancangan <i>Human Machine Interfaces</i>	31
3.4	Integrasi <i>Plant</i> , Sistem Komunikasi, dan <i>Human Machine Interface</i>	32
3.5	Perancangan dan Implementasi Sistem Pengendalian PID	32
3.5.1	Uji Open Loop Sistem	33
3.5.2	Perancangan dan Implementasi Sistem Pengendalian PID	36
3.6	Perancangan dan Implementasi <i>Neural Network</i>	36
3.6.1	Perancangan Arsitektur <i>Neural Network</i>	37
3.6.2	Variabel <i>Input Output Neural Network</i>	39
3.6.3	Inisialisasi Bobot <i>Neural Network</i>	41
BAB IV HASIL DAN PEMBAHASAN		45
4.1	Analisis Hasil Pengujian Sistem Komunikasi <i>Wireless Sensor and Actuator Networks</i>	45
4.2	Analisis Performansi Sistem Pengendalian PID	46
4.2.1	Sistem Pengendalian PID Level Campuran.....	46
4.2.2	Sistem Pengendalian PID Level Minyak	49
4.3	Analisis Performansi Sistem Pengendalian <i>Neural Network</i> PID.....	51

4.3.1	Sistem Pengendalian <i>Neural Network</i> PID Level Campuran.....	51
4.3.2	Sistem Pengendalian <i>Neural Network</i> PID Level Minyak.....	52
4.4	Analisis Perbandingan Sistem Pengendalian PID dan <i>Neural Network</i> PID.....	53
4.4.1	Perbandingan Hasil Uji Level Campuran.....	53
4.4.2	Perbandingan Hasil Uji Level Minyak.....	55
BAB V KESIMPULAN DAN SARAN		57
5.1	Kesimpulan	57
5.2	Saran.....	57
DAFTAR PUSTAKA		59
LAMPIRAN.....		lxiii
BIODATA PENULIS		lxxiii

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2. 1 P&ID Separator Tiga Fasa (Oilprocessing, 2019).....	5
Gambar 2. 2 Arsitektur IoT yang Umum Digunakan (Lombardi <i>et al.</i> , 2021)	6
Gambar 2. 3 Elemen Pada IoT (Al-Fuqaha et al., 2015).....	7
Gambar 2. 4 Skema Firebase dengan REST API (Diligent Dev, 2020)	9
Gambar 2. 5 Diagram Blok Kontroler PID (Chairuzzaini <i>et al.</i> , 1998)	10
Gambar 2. 6 Pengukuran untuk Tuning Level Control (Dataforth Corporation, 2020).....	12
Gambar 2. 7 Analisis Pada Respon Dinamik Sistem (Ogata, 2010)	14
Gambar 2. 8 Arsitektur Multilayer Perceptron (Fausett, 1993)	15
Gambar 2. 9 Alur <i>Backpropagation</i> Pada 3 Layer <i>Neural Network</i> (Negnevitsky, 2005) 17	
Gambar 2. 10 Mikrokontroler ESP32 dan <i>Pinout Diagram</i> (Circuits4you, 2018)	19
Gambar 2. 11 Sensor MS5611 dan <i>Pinout Diagram</i> (Components101, 2021).....	20
Gambar 2. 12 Bagian Pada Servo Motor (Dejan, 2020)	21
Gambar 2. 13 DC Motor Servo Valve (Parker, 2016)	21
Gambar 2. 14 Modul Komunikasi LoRa RFM95 Beserta <i>Pinout Diagram</i> (HOPERF, 2020)	22
Gambar 3. 1 Diagram Alir Metode Penelitian	23
Gambar 3. 2 Skema Separator	24
Gambar 3. 3 Desain Tabung Separator	25
Gambar 3. 4 <i>Mini Plant</i> Separator	26
Gambar 3. 5 Rancangan Sistem Elektrikal.....	27
Gambar 3. 6 Hasil Rancangan dan Realisasi Papan Elektrik Sistem Kontroler <i>Slave</i>	27
Gambar 3. 7 Rangkaian Keseluruhan Sistem Elektrikal <i>Plant</i>	28
Gambar 3. 8 Rangkaian Kontroler <i>Master</i>	28
Gambar 3. 9 Skema Sistem Komunikasi <i>Plant</i>	30
Gambar 3. 10 Tampilan HMI.....	31
Gambar 3. 11 Diagram Blok Sistem Pengendalian PID	32
Gambar 3. 12 Respon Open Loop Campuran	33
Gambar 3. 13 Respon Open Loop Minyak	34
Gambar 3. 14 Diagram Blok Sistem Pengendalian <i>Neural Network</i> PID.....	37
Gambar 3. 15 Struktur Perancangan <i>Neural Network</i> pada Pengendali <i>Neural Network</i> PID	38
Gambar 4. 1 Lokasi Pengambilan Data Sistem Komunikasi	45

Gambar 4. 2 Respon Dinamik Sistem Pengendalian PID Level Campuran.....	47
Gambar 4. 3 Respon Dinamik Sistem Pengendalian PID Level Campuran Setelah <i>Fine Tuning</i>	48
Gambar 4. 4 Respon Dinamik Pengendali PID Level Minyak	49
Gambar 4. 5 Respon Dinamik Sistem Pengendalian PID Level Minyak Setelah <i>Fine Tuning</i>	50
Gambar 4. 6 Respon Dinamik Sistem Pengendalian NN-PID Level Campuran	52
Gambar 4. 7 Respon Dinamik Sistem Pengendalian NN-PID Level Minyak.....	53
Gambar 4. 8 Perbandingan Respon Sistem Pengendalian PID dan <i>Neural Network</i> PID Pada Level Campuran.....	54
Gambar 4. 9 Perbandingan Respon Sistem Pengendalian PID dan <i>Neural Network</i> PID Pada Level Minyak	55

DAFTAR TABEL

Tabel 2. 1 Aturan Metode Level Control Loop (Dataforth Corporation, 2020)	13
Tabel 2. 2 Efek Parameter PID pada <i>closed-loop</i> response (Li <i>et al.</i> , 2006)	13
Tabel 2. 3 Spesifikasi LoRa RFM95 (HOPERF, 2020).....	22
Tabel 3. 1 Dimensi Mini Plant Separator	25
Tabel 3. 2 Penyesuaian Modul Sensor Untuk Pengukuran Level	29
Tabel 3. 3 Parameter Respon Open Loop Campuran.....	33
Tabel 3. 4 Parameter Respon Open Loop Minyak.....	35
Tabel 3. 5 Hasil Parameter PID dari Pengujian Open Loop	36
Tabel 3. 6 Parameter Model JST	39
Tabel 3. 7 Masukan <i>Error</i>	40
Tabel 3. 8 Masukan <i>Delta Error</i>	40
Tabel 3. 9 Keluaran Parameter PID	40
Tabel 3. 10 Tabel Kombinasi Parameter PID Level Campuran.....	41
Tabel 3. 11 Tabel Kombinasi Parameter PID Level Minyak.....	41
Tabel 3. 12 Jumlah Iterasi dan MSE Setelah Pelatihan Selesai	42
Tabel 3. 13 Bobot antara <i>input layer</i> dan <i>hidden layer</i> Level Campuran	42
Tabel 3. 14 Bobot antara <i>hidden layer</i> dan <i>output layer</i> Level Campuran	43
Tabel 3. 15 Bobot antara <i>input layer</i> dan <i>hidden layer</i> Level Minyak	43
Tabel 3. 16 Bobot antara <i>input layer</i> dan <i>hidden layer</i> Level Minyak	43
Tabel 4. 1 Performansi Komunikasi Wireless.....	46
Tabel 4. 2 Respon Dinamik Level Campuran dengan Sistem Pengendalian PID	47
Tabel 4. 3 Respon Dinamik Level Minyak dengan Sistem Pengendalian PID Setelah <i>Fine Tuning</i>	48
Tabel 4. 4 Respon Dinamik Sistem Pengendalian PID Level Campuran dengan Variasi Kombinasi Kp, Ki, dan Kd	49
Tabel 4. 5 Respon Dinamik Level Minyak dengan Pengendali PID	50
Tabel 4. 6 Respon Dinamik Level Minyak dengan Sistem Pengendalian PID Setelah <i>Fine Tuning</i>	51
Tabel 4. 7 Respon Dinamik Sistem Pengendalian PID Level Minyak dengan Variasi Kombinasi Kp, Ki, dan Kd	51
Tabel 4. 8 Respon Dinamik Level Campuran dengan Sistem Pengendalian <i>Neural Network</i>	52

Tabel 4. 9 Respon Dinamik Level Minyak dengan Sistem Pengendali <i>Neural Network</i> PID	53
Tabel 4. 10 Perbandingan Respon Dinamik Sistem Pengendalian PID dan <i>Neural Network</i>	54
Tabel 4. 11 Perbandingan Respon Dinamik Sistem Pengendalian PID dan <i>Neural Network</i>	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada sektor industri oil & gas, terdapat satu proses yang terjadi di industri oil & gas ialah adanya separasi antara minyak, air, dan gas yang terjadi di instrument separator 3 fasa. Secara fungsional, 3 phase separator ini dapat dikontrol level pengeluaran air, minyak dan gas serta dapat diatur debit masuk dan debit keluar. Pada Separator PV 9900 PT. JOB Pertamina-Petrochina Tuban, pengendalian level seperti minyak pada separator sangat dibutuhkan dikarenakan beberapa hal. Yang pertama, agar minyak tidak sampai terbawa oleh aliran keluaran gas (gas outlet) atau carry over yang dapat menyebabkan beratnya kerja scrubber. Kemudian, agar minyak tidak sampai masuk ke aliran keluaran air (water outlet) yang bisa menyebabkan berkurangnya produksi minyak dan terganggunya proses air yang dialirkan ke plant Sulfure Recovery Unit (SRU). Dan yang terakhir, agar produksi minyak stabil dan lebih optimal (Syauqi dan Suyanto, 2011).

Pada sistem pengendalian, metode yang banyak digunakan kebanyakan industri saat ini adalah PID. Sistem pengendalian PID digunakan 97% dari sebelas ribu pengontrol pada industri pemurnian, kimia, dan dan kertas (Gude *et al.*, 2006). Pada instrumen separator PT. Pertamina EP, kontrol proses yang digunakan masih banyak menggunakan PID klasik yang harus secara kontinu dimonitor oleh manusia secara 24 jam sehari, kadang juga dengan sistem kendali manual seperti ini menyebabkan banyak faktor dalam perhitungan kesalahan data logging harian (Ismail *et al.*, 2021). PID klasik akan sulit mencapai target yang memuaskan karena parameter PID butuh dioptimisasi (Wei, 2018). Dalam penelitian lainnya, digunakan metode kontrol PID neural-network controller untuk mengontrol pemanas, hasil tersebut menunjukkan performansi yang baik dengan penggunaan PID berbasis neural network dibandingkan dengan fuzzy PID (Liang *et al.*, 2021). Sistem auto-tuning PID dibandingkan antara kontrol PID dengan berbagai algoritma terhadap PID konvensional karena PID konvensional tidak berlaku dalam kondisi sistem yang non-linier (Yang, 2020), performansi PID dengan dasar neural network lebih baik dibandingkan fuzzy PID dan PID konvensional.

Pada saat ini, berbagai perusahaan masih menggunakan framework SCADA untuk melakukan kontrol dan monitoring dan terintegrasi dengan PLC. SCADA kurang cocok dengan posisi tersebut karena memiliki keterbatasan seperti tidak terukur karena pemrosesan

data yang lambat dengan penyimpanan memori yang minimum (Priyanka et al., 2020). Dengan pertumbuhan revolusi IoT, sistem SCADA bergerak ke generasi berikutnya, di mana ia dapat memanfaatkan teknologi cloud dan menjadi semakin *real-time*. Penggunaan protokol jaringan terbuka yang melekat pada teknologi IoT menyediakan mekanisme yang lebih baik dan andal untuk mengelola implementasi SCADA yang terdesentralisasi (Geng, 2017). Internet of Things (IoT) memberikan peluang yang menjanjikan untuk membangun sistem pada industri dengan memanfaatkan jaringan nirkabel dan perangkat sensor. Berbagai bidang industri masih terus melakukan pengembangan IoT karena mampu mendatangkan keuntungan ekonomi secara signifikan. Teknologi nirkabel berbasis IoT merupakan solusi untuk permasalahan tersebut sehingga operator mampu memonitor dan mengontrol proses produksi minyak dari jarak jauh untuk menekan penggunaan biaya dan mengurangi tingkat kecelakaan kerja.

Pada Penelitian sebelumnya penelitian sebelumnya telah dilakukan rancang bangun mini plant separator tiga fasa berbasis database IoT dengan monitoring secara *real-time* melalui DCS. Penelitian menggunakan metode kendali berbasis *fuzzy* PI dengan *rule base* nya *input error* (Sudrajat, 2020). Kemudian selanjutnya dikembangkan sistem kontrol dan monitoring berbasis *Industrial Internet of Things* pada mini plant separator tiga fasa dengan menggunakan *real-time* database dan interface berbasis web yang menggunakan metode kendali *fuzzy* PID dengan *input error* dan *change error* (Subakti, 2021). Kedua penelitian di atas menggunakan sistem komunikasi yang berbasis protokol IEEE 802.15.4 dengan modul ZigBee.

Berdasarkan uraian diatas, maka pada penelitian ini akan dilakukan perancangan sistem pengendalian dengan metode Neural Network PID dan sistem monitoring dan kontrol berbasis Internet of Things. Algoritma Neural Network ini digunakan untuk mempelajari perubahan beban objek mini plant dan melakukan tuning parameter PID secara adaptif selama proses berlangsung sehingga dapat memenuhi set point yang diinginkan. Sistem pengendali yang diterapkan dapat dipantau dan dikendalikan dengan jarak yang jauh dengan kombinasi sistem komunikasi berbasis frekuensi dengan modul LoRa yang diintegrasikan dengan *cloud database* secara *realtime*, LoRa memiliki *coverage area* lebih jauh dibandingkan dengan ZigBee serta hemat daya dengan *bitrate* yang cukup untuk mengirim seluruh parameter yang dibutuhkan dalam monitoring dan kontrol, sehingga operator dapat mengendalikan proses yang terjadi dengan jarak yang jauh.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah pada penelitian ini adalah:

- a. Apakah sistem pengendalian Neural Network PID dapat meningkatkan performansi sistem pengendalian level pada mini plant separator tiga fasa dibandingkan dengan sistem pengendalian PID konvensional?
- b. Apakah sistem komunikasi jarak jauh berbasis Internet of Things pada mini plant separator tiga fasa dapat berkomunikasi secara *real-time*?

1.3 Tujuan

Tujuan dilakukannya penelitian ini adalah:

- a. Menganalisis performansi apakah sistem pengendalian Neural Network PID dapat meningkatkan performansi sistem pengendalian level pada mini plant separator tiga fasa dibandingkan dengan sistem pengendalian PID konvensional.
- b. Menganalisis performansi apakah sistem komunikasi jarak jauh berbasis Internet of Things pada mini plant separator tiga fasa dapat berkomunikasi secara *real-time*.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

- a) Data yang digunakan dan data pengukuran terbatas pada skala laboratorium.
- b) Fasa yang digunakan adalah fasa air dan minyak.
- c) Karakteristik Control Valve dianggap linier.
- d) Laju aliran *input* konstan.
- e) Pengujian dilakukan selama 1300 detik untuk menghindari kekosongan tangki *supply*
- f) Sistem Komunikasi *plant* dilakukan secara Point-to-Point (P2P) dengan kondisi Line of Sight (LoS)

1.5 Sistematika Laporan

Sistematika penulisan laporan tugas akhir secara garis besar ialah sebagai berikut:

a. BAB I Pendahuluan

Bab I terdiri dari latar belakang, perumusan masalah, tujuan penelitian, batasan masalah dan sistematika laporan.

b. BAB II Tinjauan Pustaka dan Dasar Teori

Pada Bab II dibahas mengenai teori-teori yang berkaitan dengan penelitian yang akan dilakukan seperti Separasi Minyak Tiga Fasa, Internet of Things, Industrial

Wireless Sensor and Actuator Networks, Firebase Realtime Database, Sistem Pengendalian PID, Analisis Respon Dinamik Sistem, Jaringan Syaraf Tiruan, Mikrokontroler ESP32, Sensor MS5611, Servo Motor Control Valve, dan Transceiver LoRa RFM95.

c. BAB III Metodologi Penelitian

Bab III berisi mengenai diagram alir dari penelitian yang dilakukan, metode, dan langkah-langkah yang dilakukan dalam penelitian.

d. BAB IV Hasil dan Pembahasan

Bab IV berisi hasil penelitian yang dilakukan seperti analisis komunikasi *Wireless Sensor and Actuator Networks*, Uji performansi sistem pengendalian PID, *Neural Network* PID, dan perbandingan masing-masing performansi sistem pengendali.

e. BAB V Kesimpulan dan Saran

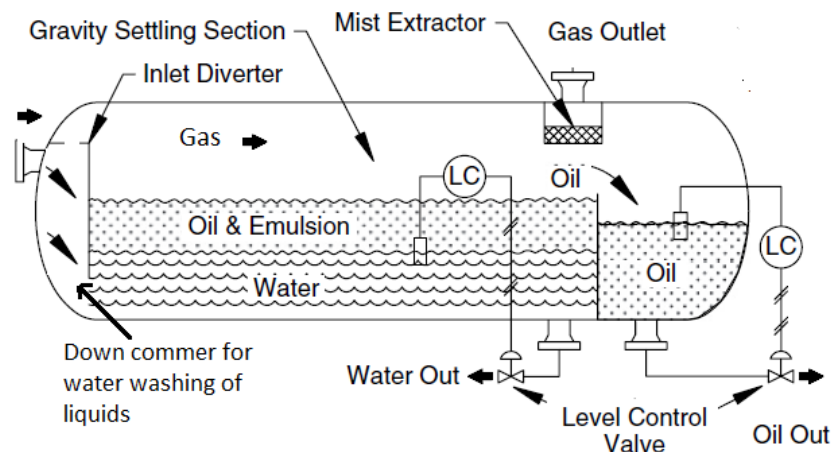
Bab V diberikan kesimpulan mengenai rumusan masalah pada tugas akhir ini serta saran sebagai penunjang maupun pengembangan pada tugas akhir selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Proses Separasi Minyak Tiga Fasa

Proses Separasi minyak adalah proses utama di fasilitas produksi petroleum dimana fluida dari sumur pengeboran dipisahkan komponennya. Terdapat 2 jenis separator yaitu dua dan tiga fasa. Pada tiga fasa, fluida dari sumur pengeboran dipisahkan menjadi air, minyak dan gas (Ghaffarkhah et al., 2017). Proses separasi umumnya mengandalkan konsep gravitasi dan massa jenis. Inlet yang berasal dari aliran sumur dapat terdiri dari minyak mentah, gas, kondensat, air dan berbagai kontaminan. Tujuan dari separator adalah untuk membagi aliran menjadi fraksi yang diinginkan. Pada *production separator*, tekanan separator berkisar antara 3-5 MPa dengan suhu pada inlet berkisar pada 100 °C – 150 °C (Havard, 2013).



Gambar 2. 1 P&ID Separator Tiga Fasa (Oilprocessing, 2019)

Berdasarkan Gambar 2. 1, proses separasi dilakukan setelah fluida melewati *production wellhead* yang kemudian terbagi menuju *production and test manifolds* serta *production separator* yang dikontrol level dan tekanannya. Dalam proses separasi minyak, terjadi masalah seperti *overpressure* dan *overflow*. *Overpressure* disebabkan oleh tekanan gas yang melebihi batas kemampuan yang dapat ditampung tabung separator. Masalah ini diatasi dengan melepaskan gas ke flare. *Overflow* disebabkan air yang akan dipisahkan dari minyak meluap dan tercampur kembali dengan minyak. Hal ini dapat diatasi dengan mengendalikan aliran masuk dan aliran keluar tabung separator. Pengontrolan yang dilakukan pada proses pemisahan ini adalah pengaturan flow inlet dan outlet, level serta tekanan pada tabung separator. Pengontrolan dilakukan dengan

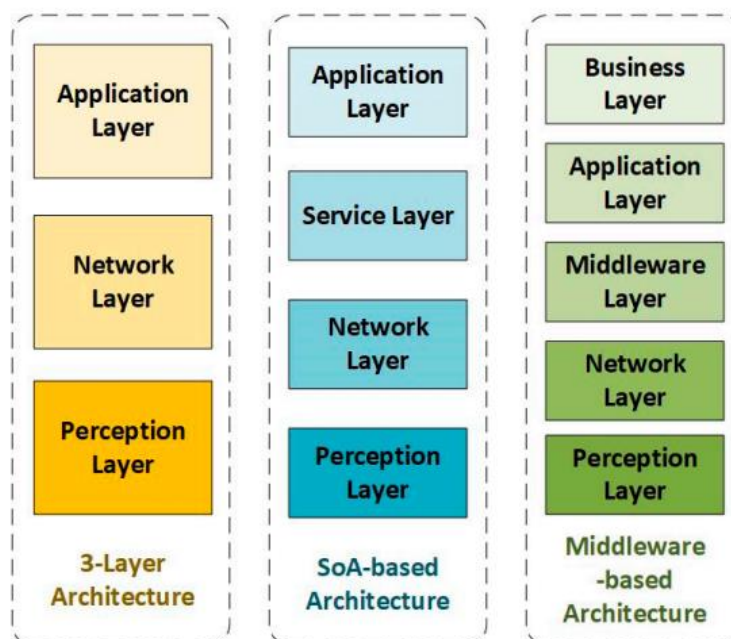
menggunakan pembacaan *level transmitter* untuk mengaktivasikan *control valve* disaat akan terjadi peristiwa *overflow*.

2.2 Internet of Things

Internet of Things (IoTs) adalah teknologi baru yang paling signifikan dalam beberapa tahun terakhir yang memiliki efek transformasional ke setiap industri di seluruh dunia. IoT adalah teknologi mendigitalisasi dunia fisik yang ada saat ini. IoT merupakan perkembangan dari teknologi sebelumnya: *pervasive information systems*, *sensor networks*, sistem komputasi tertanam (*embedded computing*) (Serpanos dan Wolf, 2017). IoT diharapkan menawarkan solusi yang menjanjikan untuk mengubah operasi dan peran banyak sistem industri yang ada seperti sistem transportasi dan sistem manufaktur.

2.2.1 Arsitektur IoT

Dalam merancang sistem berbasis IoT, perlu dirancang pula arsitektur dari IoT itu sendiri. Perancangan arsitektur IoT ditujukan untuk menjamin beroperasinya IoT, yang menjembatani antara dunia nyata dan *virtual*. Desain IoT melibatkan seperti jaringan, komunikasi, dan sistem keamanan. Sebuah arsitektur IoT harus bersifat memiliki sistem yang aman, terukur dalam peningkatan jumlah perangkat tanpa menurunkan performa sebelumnya, beroperasi dengan daya serendah mungkin, dan dapat bekerja dengan baik dengan sistem lain atau *interoperable* (Lombardi *et al.*, 2021).



Gambar 2. 2 Arsitektur IoT yang Umum Digunakan (Lombardi *et al.*, 2021)

Dengan kondisi saat ini yang tidak adanya referensi tunggal terkait arsitektur IoT serta upaya standarisasi, Gambar 2. 2 merupakan arsitektur IoT yang paling banyak digunakan. Dalam penelitian ini digunakan arsitektur yang paling sederhana yaitu arsitektur 3 lapis. *Perception layer* merepresentasikan lapisan fisik objek dan mengelompokkan semua fitur dengan mengumpulkan dan memproses informasi. Lapisan ini berada pada objek yang diintegrasikan IoT sehingga mencakup mikrokontroler, sensor, dan sistem komunikasi. *Network layer* merepresentasikan lapisan komunikasi yang bertanggung jawab untuk transmisi data ke lapisan aplikasi melalui berbagai teknologi dan protokol. Contoh protokol yang digunakan pada ranah IoT ialah IEEE 802.15.4 (ZigBee), IEEE 802.11 (WiFi), IEEE 802.15.1 (Bluetooth), IEEE 802.3 (Ethernet), LPWAN (LoRaWAN), IEEE 1901 (PLC), RFID, NFC, dan Z-Wave. *Application layer* merepresentasikan semua perangkat lunak yang diperlukan untuk layanan tertentu. Data dari level sebelumnya disimpan dan diproses digunakan. Data hasil dari proses pemrosesan disediakan untuk aplikasi IoT. Salah satu penggunaan untuk mengelola data ialah cloud computing. Contoh platform cloud yang digunakan untuk aplikasi IoT ialah Amazon AWS, Microsoft Azure, Xively, dan Firefox WebThings Gateway (Lombardi *et al.*, 2021).

2.2.2 Elemen IoT

Dalam perancangan arsitektur IoT. Terdapat didalamnya elemen-elemen yang berperan dalam fungsionalitas IoT. Berikut merupakan enam elemen IoT yang ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Elemen Pada IoT (Al-Fuqaha *et al.*, 2015)

Dalam elemen IoT, *identification* pada IoT untuk memberi nama dan mencocokkan layanan dengan permintaan. Terdapat identifikasi unik seperti ID yang mencocokkan antara objek dan alamatnya. Identifikasi digunakan untuk memberikan identitas yang jelas untuk setiap objek dalam jaringan. Elemen *sensing* berupa sensor yang data yang dikumpulkan dianalisis untuk mengambil tindakan spesifik berdasarkan layanan yang diperlukan. Elemen *communication* berperan dengan menghubungkan objek tertentu. Objek ini dihubungkan dengan protokol komunikasi tertentu. Elemen *computation* merupakan mikrokontroler pada sistem IoT. Beragam contoh mikrokontroler seperti Arduino, Intel Galileo, ESP32,

ESP8266, dan Raspberry PI. Elemen *services* merupakan elemen yang menyediakan fungsionalitas untuk mengintegrasikan layanan dan interfaces pada IoT. Elemen ini memungkinkan di mana platform perangkat keras dan perangkat lunak dapat digunakan kembali. Elemen *semantics* bertujuan mengenali dan menganalisis data untuk memahami keputusan yang tepat untuk menyediakan layanan yang tepat. Dengan demikian, semantik merepresentasikan otak IoT dengan mengirimkan permintaan ke resource yang tepat (Al-Fuqaha et al., 2015).

2.3 Industrial Wireless Sensor and Actuator Networks

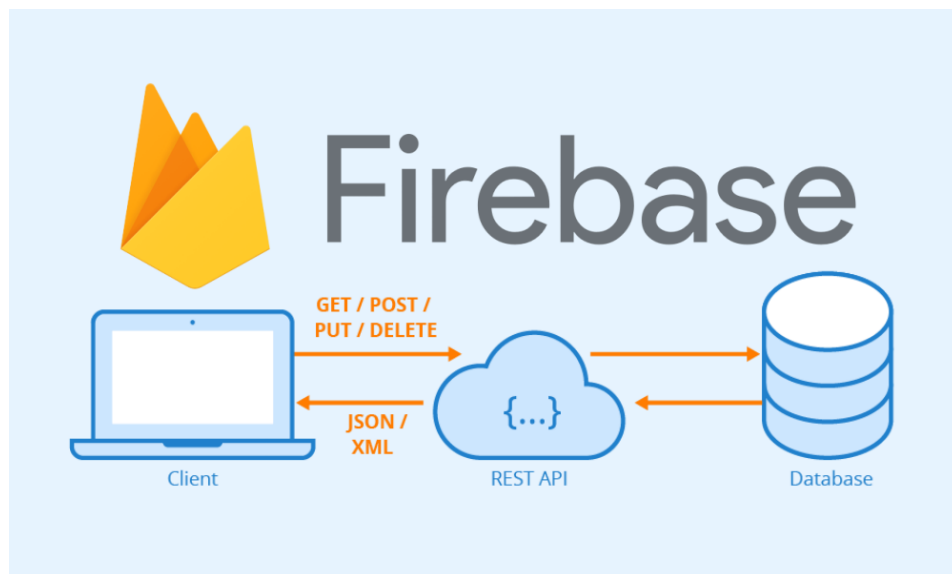
Sistem dalam industri dibutuhkan pemantauan untuk menghasilkan produksi yang tinggi. Teknologi yang dikembangkan untuk sistem monitoring lingkungan ialah *Wireless Sensor Network* (WSN). WSN terdiri atas banyak dari node sensor yang memiliki kapasitas terbatas dalam pemrosesan data. WSN hanya memonitor lingkungan yang ingin di monitor. Berbeda dengan *Wireless Sensor and Actuator Networks* (WSAN), WSAN berperan dapat mengontrol yang dimana sensor berperan sebagai fungsi untuk menggerakkan actuator (Chen et al., 2011).

WSAN memiliki fleksibilitas yang tinggi dengan biaya instalasi yang rendah dibandingkan tradisional SCADA (Grilo et al., 2014). Dalam banyak skenario aplikasi, jaringan nirkabel IEEE 802.15.4 digunakan untuk menghubungkan objek pintar yang dibatasi konsumsi energinya ke internet, karena menyediakan konektivitas jarak pendek dengan konsumsi energi yang terbatas. IEEE 802.15.4 saat ini merupakan teknologi akses paling umum untuk IoT. Dalam beberapa tahun terakhir, IEEE telah mengeluarkan sejumlah amandemen untuk memperluas standar 802.15.4 asli, dimana Standar IEEE 802.15.4g adalah teknologi nirkabel yang berpotensi meningkatkan kemampuan Internet of Things. Standar ini dapat digunakan di sejumlah domain aplikasi potensial, baik di lingkungan perkotaan maupun pedesaan, karena menyediakan komunikasi daya rendah dan biaya rendah jarak jauh, sehingga menjadikan *Low Power Wide Area Network* (LPWA). Standar IEEE 802.15.4g dapat digunakan pada modul perangkat LoRa yang digunakan karena LPWA mengirimkan data dengan ukuran kecil untuk berkomunikasi dengan jarak yang jauh hingga beberapa kilometer (Righetti et al., 2019).

2.4 Firebase Real Time Database

Dalam sistem berbasis IoT, pemrosesan data dilakukan di *cloud* sehingga memudahkan klien dalam mengakses data. Salah satu platform Firebase merupakan platform

yang dikembangkan oleh google pada tahun 2016 untuk memenuhi kebutuhan *developer* dalam mengembangkan infrastruktur untuk membangun sebuah aplikasi ataupun sebuah website. Salah satu layanan yang digunakan dalam penelitian ini dengan platform Firebase ialah Firebase Realtime Database (RTDB). Firebase RTDB merupakan database berbasis *cloud* yang bertipe NoSQL database (Moroney, 2017). NoSQL (*Not Only Structured Query Language*) memiliki bentuk data *non-relational* yang tidak berdasarkan baris dan kolom seperti SQL yang termasuk ke dalam *relational database*, sehingga Firebase RTDB bekerja menerima data dengan format JSON.



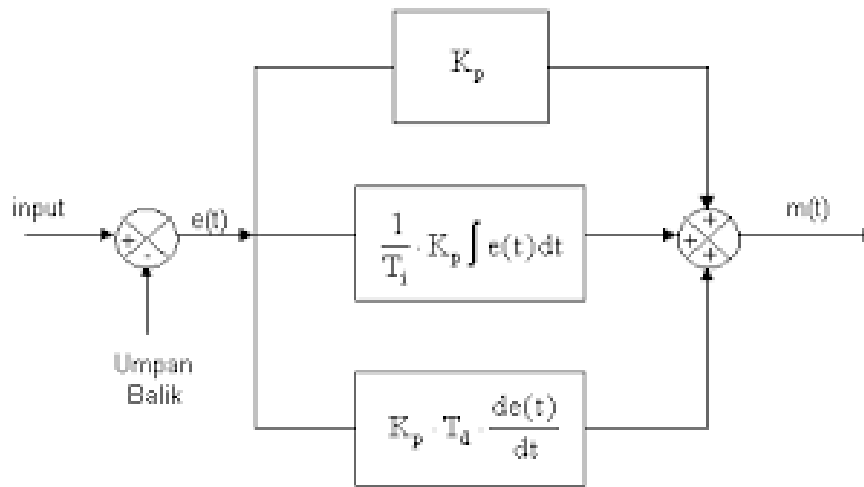
Gambar 2. 4 Skema Firebase dengan REST API (Diligent Dev, 2020)

Gambar 2.4 merupakan skema kerja firebase sebagai *cloud database*. Firebase RTDB melakukan sinkronisasi secara *real-time* kepada setiap klien sehingga klien secara otomatis menerima data terbaru. Klien terhubung dengan Firebase RTDB dengan dijumpai oleh *REST API*. Ini memungkinkan klien mengambil, mengubah, ataupun menghapus data. Pemanfaatan *REST API* ini dapat memungkinkan database dapat dibatasi dan diakses dengan platform apapun, dimana dalam penelitian ini digunakan untuk memonitor data dan mengendalikan *plant* dengan jarak jauh.

2.5 Sistem Pengendalian PID

Sistem Pengendalian PID merupakan salah satu jenis sistem pengendalian konvensional yang yang paling banyak digunakan dalam industri proses. Kesederhanaan struktur kontrol PID juga menjadi salah satu alasan mengapa pengendali ini sangat mudah diimplementasikan di mana saja. Pada sistem pengendalian Gambar 2. 5, PID terdapat 3 komponen penyusun utama. 3 komponen tersebut yaitu konstanta proporsional yang

dilambungkan dengan K_p , konstanta waktu integral dengan T_i , dan konstanta waktu differensial dengan T_d .



Gambar 2. 5 Diagram Blok Kontroler PID (Chairuzzaini *et al.*, 1998)

Dalam suatu sistem, pengendalian dilakukan dengan adanya error akibat deviasi antara *process variable* dengan *set point*. Persamaan PID diberi *input* berupa *error signal* dan output berupa *control signal* (Li *et al.*, 2006). Persamaan (2.1) merupakan persamaan umum PID yang didalamnya terdapat ketiga konstanta (*proportional*, *integral*, dan *derivative*) yang disebut dengan parameter pengendali PID. Parameter PID menentukan performa dari kontroler PID.

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{d(e(t))}{dt} \right] \quad (2.1)$$

dengan:

$u(t)$ = Sinyal kontrol

K_p = *Proportional Gain*

T_i = *Integral Time*

T_d = *Derivative Time*

$e(t)$ = Error antara *set point* dengan nilai proses aktual

Kontrol proporsional memiliki keluaran yang sama dengan error pada sistem waktu itu juga. Output akan besar saat error pada sistem besar, begitu pula sebaliknya Aksi kontrol proporsional akan menghasilkan *steady-state error*. Pada integral, aksi integral memberikan output berdasarkan *cumulative error* dari kesalahan yang terjadi sebelumnya. Akumulasi dari *error* kemudian dikalikan dengan *gain* dari *integral*. Aksi integral ini dapat mengeliminasi *steady-state error* menuju nol. Sedangkan kontrol derivative bekerja terhadap laju perubahan *error* sistem kendali. Kendali ini akan bergantung terhadap

tren/*slope* dari kesalahan sistem, sehingga apabila *error* terjadi dengan frekuensi yang tinggi, derivatif cenderung menghasilkan output yang besar (Goodwin et al., 2000).

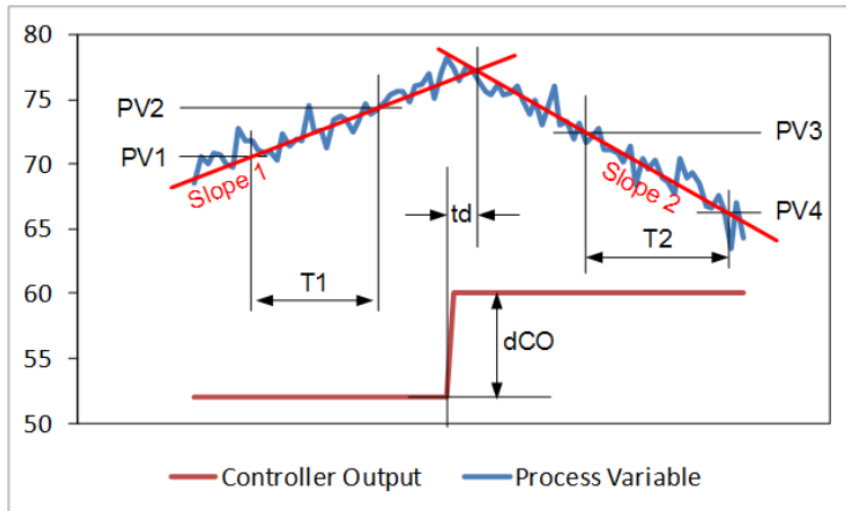
Dalam praktiknya, untuk memenuhi spesifikasi sistem pengaturan yang diharapkan dengan *set point* yang telah ditetapkan, parameter PID harus *tuning* terlebih dahulu. Ada berbagai macam metode *tuning* nilai parameter PID, yaitu *Direct Synthesis*, Ziegler Nichols, Cohen-Coon, dan lain sebagainya.

2.5.1 Tuning Parameter PID untuk Level Control

Proses pemilihan parameter untuk memenuhi performa dari spesifikasi suatu sistem diperlukan *tuning* sistem kontrol. Salah satu metode *tuning* yang banyak digunakan pada industri ialah metode Ziegler Nichols. Metode ini berguna ketiga pemodelan matematis dari *plant* tidak diketahui. Aturan dari metode ini dapat menentukan nilai K_p , T_i , dan T_d berdasarkan karakteristik pada respon transien dari suatu *plant* (Ogata, 2010).

Dalam pengendaliannya, sistem pengendalian *flow*, *pressure*, dan *temperature* dibedakan dengan *level control*. Metode Ziegler Nichols harus dimodifikasi karena perilaku proses *control level* merupakan *non-self regulating process* atau *integrating process*. Dimana objek sistem tidak memiliki titik kesetimbangan. Level akan terus bergerak sampai operator proses yang campur tangan (King, 2011).

Proses tuning PID dilakukan dengan pengujian *open loop*, uji *open loop* dilakukan tanpa pengontrol atau, jika ada, dengan pengontrol dalam mode manual. Gangguan disuntikkan ke dalam proses dengan mengubah *manipulated variable* secara langsung. Aliran masuk dan keluar tidak dikontrol dan kondisi kontrol berada dalam kondisi manual. Pembacaan level akan naik dan terbentuk *slope response* naik. Kemudian diberi *step response* sampai membentuk arah *slope response* baru yang nantinya kontroler diatur kembali menuju ke titik yang sesuai, maka akan terjadi *slope response* turun.



Gambar 2. 6 Pengukuran untuk Tuning Level Control (Dataforth Corporation, 2020)

Gambar 2.6 merupakan hasil dari pengukuran membentuk karakteristik dari suatu *plant*. Titik pada *Slope 1* (PV1 dan PV2) dan *Slope 2* (PV3 dan PV4) akan digunakan untuk mendapatkan *differential slopes* (DS) persamaan (2.2).

$$DS = \frac{PV4 - PV3}{T2} - \frac{PV2 - PV1}{T1} \quad (2.2)$$

dengan:

DS = Perbedaan gradien *Slope 1* dan *Slope 2*

$PV4$ = Titik ke-2 pada *Slope 2* (cm)

$PV3$ = Titik ke-1 pada *Slope 2* (cm)

$T2$ = Waktu tempuh titik 1 dan 2 pada *slope 2* (menit)

$PV4$ = Titik ke-2 pada *Slope 1* (cm)

$PV3$ = Titik ke-1 pada *Slope 1* (cm)

$T2$ = Waktu tempuh titik 1 dan 2 pada *slope 1* (menit)

Semakin jauh antar titik yang diambil pada salah satu *slope*, maka semakin akurat pengukuran yang dilakukan. Apabila PV tidak dalam range 0-100%, maka DS dinormalisasi dan dikonversi dengan persamaan (2.3).

$$DS (\%) = 100 \times \frac{DS}{(PV \text{ range max} - PV \text{ range min})} \quad (2.3)$$

dengan:

$PV \text{ range max}$ = variabel proses tertinggi (cm)

$PV \text{ range min}$ = variabel proses terendah (cm)

Selanjutnya, dibutuhkan parameter kedua dalam tuning PID level control. Maka dilakukan perhitungan *integration rate* (r_i) dengan persamaan (2.4)

$$r_i = \frac{DS[\%]}{dCO[\%]} \quad (2.4)$$

dengan:

dCO = Perubahan Output Kontroler (%)

r_i = *Integration Rate*

Hasil dari persamaan (2.4) dapat menentukan parameter PID berdasarkan Tabel 2.1.

Tabel 2. 1 Aturan Metode Level Control Loop (Dataforth Corporation, 2020)

Parameter PID	PI	PID
K_p	$\frac{0.45}{r_i \times t_d}$	$\frac{0.45}{r_i \times t_d}$
T_i	$6.67 \times t_d$	$5 \times t_d$
T_d	0	$0.4 \times t_d$

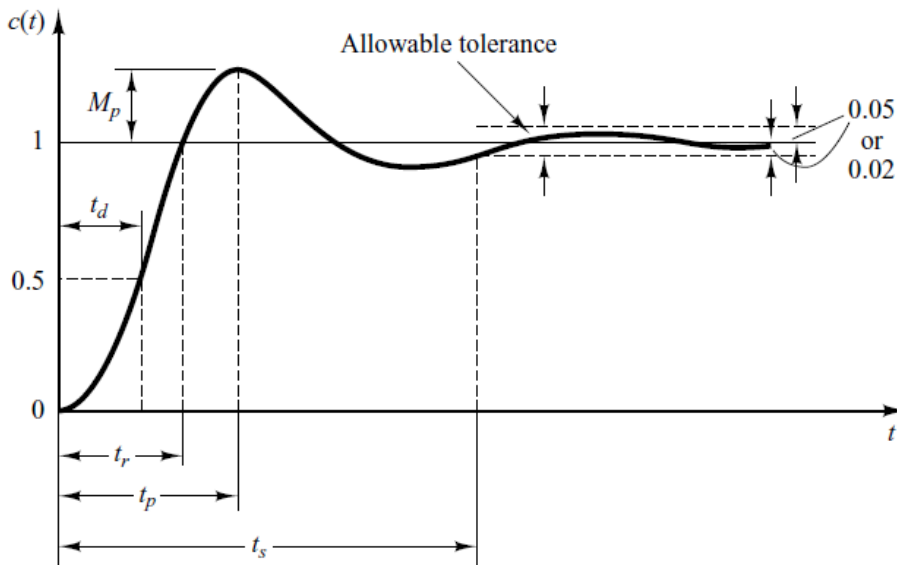
Dimana *dead time* (t_d) merupakan waktu yang dibutuhkan sistem untuk merespon variabel proses setelah diberikan perubahan sinyal kontrol. Hasil dari tuning akan disubstitusi ke persamaan umum PID (2.1). Apabila respons dari tuning PID masih terdapat overshoot yang tinggi dan steady-state error yang cukup besar, maka dapat dilakukan dengan *fine tuning* sebagai finalisasi jika diperlukan. *Fine tuning* merupakan proses mengubah parameter PID secara manual dengan tujuan agar mendapat performansi sistem yang lebih baik lagi. Naik turunnya rise time, overshoot, settling time, dan steady-state error pada hasil respon dinamik sistem. Perubahan performansi respon dinamik saat perubahan tiap parameter PID saat melakukan *fine tuning* berdasarkan Tabel 2. 2.

Tabel 2. 2 Efek Parameter PID pada *closed-loop* response (Li et al., 2006)

Parameter PID	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
K_P	Berkurang	Bertambah	Sedikit Bertambah	Berkurang	Menurun
K_I	Sedikit Berkurang	Bertambah	Bertambah	Sangat Berkurang	Menurun
K_D	Sedikit Berkurang	Berkurang	Berkurang	Sedikit Perubahan	Meningkat

2.6 Analisis Respon Dinamik Sistem

Kinerja sistem pengendalian ditentukan bagaimana respon dinamik suatu sistem saat diberi *step input*. Respon transien pada suatu pengendali akan beresilasi terlebih dahulu sebelum mencapai kondisi *steady state*. Respon transien dari berbagai sistem dapat dengan mudah dibandingkan apabila menggunakan kondisi standar yang sama yang awalnya diam dengan seluruh turunan waktu nol.



Gambar 2.7 Analisis Pada Respon Dinamik Sistem (Ogata, 2010)

Berdasarkan Gambar 2.7, karakteristik respon dinamik dari sebuah sistem pengendalian ialah sebagai berikut:

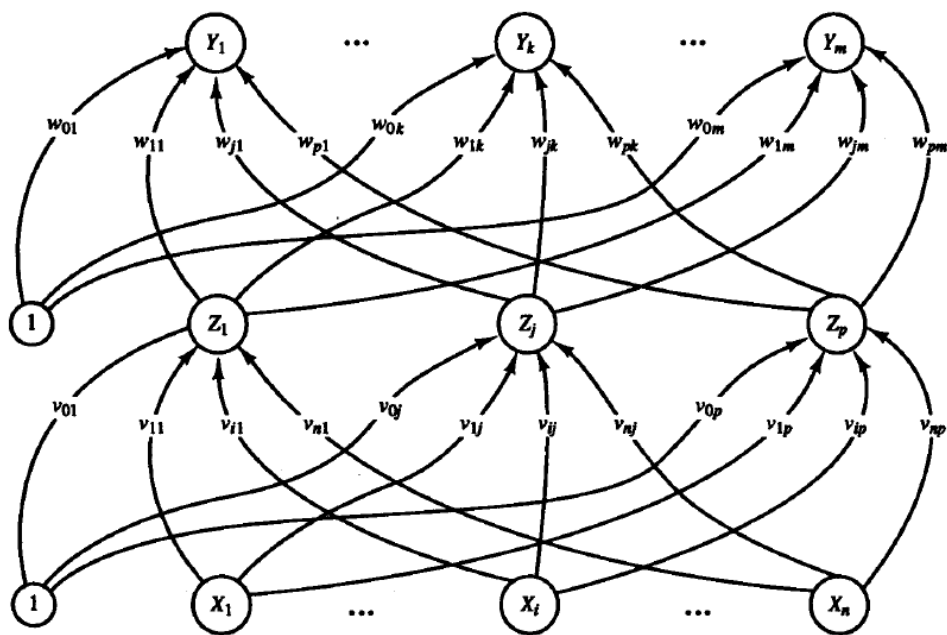
- *Rise Time* (t_d), waktu yang diperlukan untuk respon naik dari 10% ke 90%, 5% ke 95% atau 0% ke 100% berdasarkan nilai *setpoint*.
- *Maximum Overshoot* (M_p), dinyatakan dalam persen, merupakan nilai tertinggi respon yang diukur dari *setpoint*.
- *Settling Time* (t_s), waktu yang dibutuhkan respon untuk mencapai kemudian bertahan pada error terhadap *setpoint* sejauh 2% atau 5%.
- *Error Steady State* (E_{ss}), selisih antara *process variable* dengan *setpoint* saat mencapai kondisi tunak. E_{ss} dinyatakan dalam satuan persentase.

2.7 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

Jaringan Syaraf Tiruan (JST) merupakan modul komputasi yang meniru fungsi sistem syaraf biologis manusia. JST memiliki *processing elements* yang terkoneksi antara satu

dengan lainnya dengan *weighted connections*. Analogi pada sistem syaraf manusia, *processing elements* merupakan neuron yang terdiri dari *input* dari dendrit dan *output* yang merupakan sinapsis via akson (Kukreja et al., 2016). Pemodelan model neural network memiliki elemen dasar, yaitu sinapsis yang dicirikan dengan setiap bobot masing-masing, *adder* untuk menjumlahkan *input* sinyal, dan fungsi aktivasi.

Arsitektur dasar JST terbagi atas 3 bagian yang terdiri atas *input layer*, *hidden layer*, dan *output layer*. *Input layer* bertanggung jawab atas penerimaan informasi berupa data, sinyal, atau pengukuran dari lingkungan eksternal. *Hidden layer* merupakan lapisan yang terdiri dari neuron yang bertanggung jawab untuk mengekstraksi pola yang terkait dengan proses atau sistem yang dianalisis. Pada lapisan ini banyak berperan dalam terjadinya pemrosesan internal dari jaringan. *Output layer* merupakan lapisan yang menghasilkan hasil akhir dari jaringan. Lapisan ini merupakan hasil dari lapisan sebelumnya. *Output Layer* terdiri hanya satu lapisan aja (da Silva et al., 2017). Terdapat beberapa jenis arsitektur JST dengan mempertimbangkan disposisi neuron, serta bagaimana mereka saling berhubungan dan bagaimana lapisannya tersusun. Multilayer Perceptron (MLP) salah satu dari jenis arsitektur JST yang menggunakan *hidden layer*.



Gambar 2. 8 Arsitektur Multilayer Perceptron (Fausett, 1993)

Berdasarkan arsitektur MLP pada Gambar 2.8, *input node* disimbolkan sebagai X_i , *hidden node* disimbolkan sebagai Z_j , dan *output node* disimbolkan sebagai Y_k , serta i, j , dan k sesuai dengan jumlah node yang digunakan pada setiap *layer*. Persamaan untuk mendapatkan sinyal masukan menuju tiap node pada *hidden layer* berdasarkan persamaan (2.5)

$$z_{in.j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.5)$$

dengan:

$z_{in.j}$ = Input menuju setiap node *hidden layer* ($j = 1,2,3, \dots, p$)

v_{0j} = bias pada *hidden layer*

x_i = Input setiap node dari *input layer* ($i = 1,2,3, \dots, n$)

v_{ij} = Bobot dari setiap node *input layer* menuju *hidden layer*

Sinyal keluaran dari *hidden layer* diperoleh dengan menggunakan fungsi aktivasi. Fungsi aktivasi untuk membatasi amplitudo output neuron. Fungsi ini menekan (membatasi) rentang amplitudo yang diizinkan dari sinyal output ke *finite value* (Haykin, 1999). Sehingga keluaran dari setiap node pada *hidden layer* menggunakan persamaan (2.6).

$$z_j = f(z_{in.j}) \quad (2.6)$$

dengan z_j merupakan sinyal keluaran tiap node dari *hidden layer* yang menggunakan fungsi aktivasi.

Fungsi aktivasi memiliki banyak jenis seperti linear function, threshold function, dan sigmoid function. Jenis yang banyak digunakan ialah menggunakan *sigmoid function*, karena memiliki sifat nonlinier. Dengan salah satu jenis fungsi aktivasi yang digunakan adalah sigmoid biner. Persamaan fungsi aktivasi beserta turunannya berdasarkan persamaan (2.7) dan (2.8).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.7)$$

$$f'(x) = f(x)[1 - (f(x))] \quad (2.8)$$

dengan:

$f(x)$ = Fungsi Aktivasi

dengan cara yang sama maka didapat persamaan untuk memperoleh nilai *input* menuju setiap node pada *output layer* dengan persamaan (2.9).

$$y_{in.k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.9)$$

dengan:

$y_{in.k}$ = Input menuju setiap node *output layer* ($k = 1,2,3, \dots, m$)

w_{0k} = bias pada *output layer*

z_j = Input setiap node dari *hidden layer* ($j = 1,2,3, \dots, p$)

w_{jk} = Bobot dari setiap node *hidden layer* menuju *output layer*

Sehingga keluaran dari setiap node pada *output layer* menggunakan persamaan (2.10).

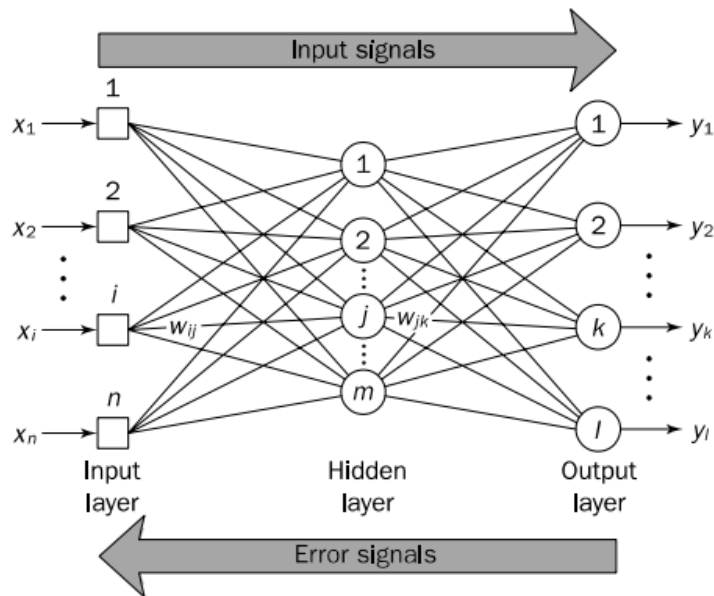
$$y_k = f(y_{in.k}) \quad (2.10)$$

dengan y_k merupakan sinyal keluaran tiap node dari *output layer* yang menggunakan fungsi aktivasi.

2.7.1 Algoritma Pelatihan *Backpropagation*

Salah satu algoritma pelatihan pada JST ialah *backpropagation*. Algoritma *backpropagation* memiliki alur seperti yang ditunjukkan pada Gambar 2.9. Secara garis besar *backpropagation* terdiri dari 3 tahapan sebagai berikut:

1. Proses umpan maju (*feed forward*)
2. Perhitungan error dari propagasi balik (*backpropagation*)
3. Penyesuaian nilai bobot berdasarkan perhitungan error (*adjustment*)



Gambar 2. 9 Alur *Backpropagation* Pada 3 Layer Neural Network (Negnevitsky, 2005)

Setelah *feedforward* dilakukan, selanjutnya dilakukan komputasi dengan membandingkan hasil dari setiap node *output layer* z_k dengan nilai target t_k untuk menentukan koreksi error δ_k . Perhitungan δ_k dilakukan untuk mendistribusikan kesalahan dari *output layer* menuju *layer* sebelumnya. Perhitungan δ_k dapat menggunakan persamaan (2.11).

$$\delta_k = (t_k - y_k)f'(y_{in.k}) \quad (2.11)$$

dengan:

δ_k = Informasi error pada *output layer*

Kemudian persamaan (2.12) untuk koreksi error ini digunakan dalam koreksi bobot antara *hidden layer* dan *output layer* untuk memperbaiki nilai w_{jk} .

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.12)$$

Sedangkan pada bias, persamaan koreksi bobot menggunakan persamaan (2.13).

$$\Delta w_{0k} = \alpha \delta_k \quad (2.13)$$

dengan:

Δw_{jk} = Koreksi bobot antara setiap node *hidden layer* dan *output layer*

Δw_{0k} = Koreksi bobot bias pada *output layer*

Langkah persamaan (2.12) diulangi sebanyak jumlah node *hidden layer*.

Kemudian pada setiap *hidden layer*, konsep yang sama digunakan dalam koreksi bobot antara *input layer* dan *hidden layer*. Setiap node pada *hidden layer* menjumlahkan *delta input* dari node yang ada di atasnya. Persamaan penjumlahan *delta input* menggunakan persamaan (2.14) dan setelah dilakukan kalkulasi dengan turunan fungsi aktivasi menggunakan persamaan (2.15).

$$\delta_{in.j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.14)$$

$$\delta_j = \delta_{in.j} f'(z_{in.j}) \quad (2.15)$$

dengan:

δ_j = Informasi error pada *hidden layer*

Kemudian dilakukan perhitungan koreksi bobot untuk memperbaiki nilai v_{ij} menggunakan persamaan (2.16).

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.16)$$

Sedangkan pada bias, persamaan koreksi bobot menggunakan persamaan (2.17).

$$\Delta v_{0j} = \alpha \delta_j \quad (2.17)$$

dengan:

Δv_{ij} = Koreksi bobot antara setiap node *input layer* dan *hidden layer*

Δv_{0j} = Koreksi bobot bias pada *hidden layer*

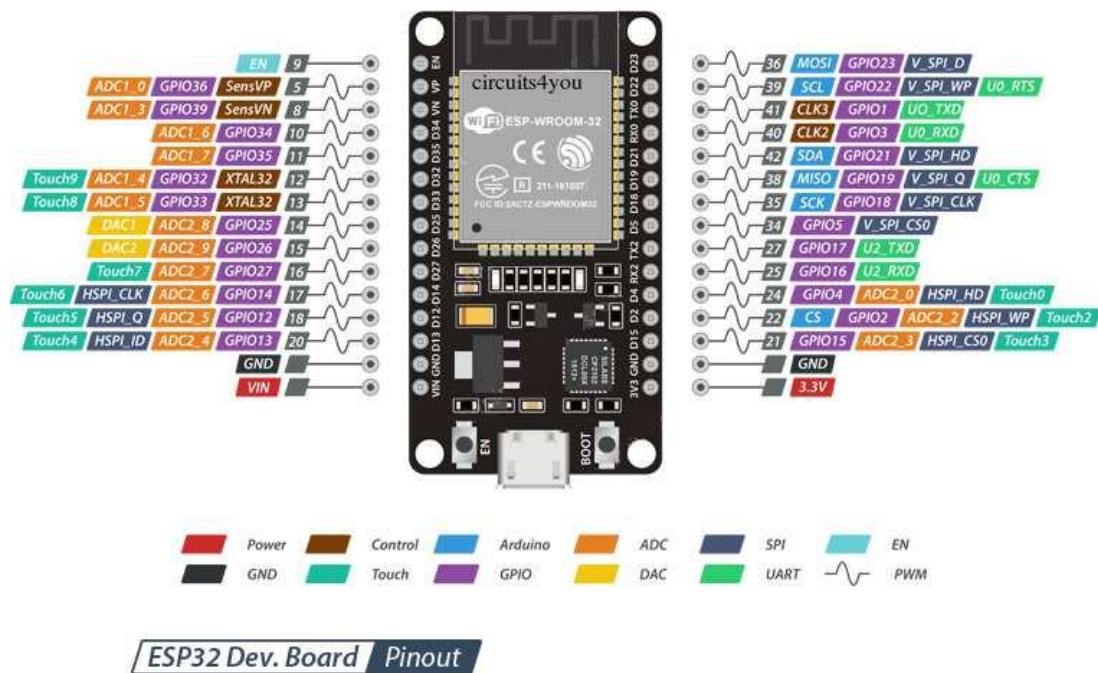
Kemudian dilakukan perbaikan pada bobot antara *input layer-hidden layer* dan *hidden layer-output layer* beserta biasnya dengan persamaan (2.18) dan (2.19).

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.18)$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.19)$$

2.8 Mikrokontroler ESP32

ESP32 merupakan chip yang dikembangkan oleh Espressif Systems dengan *low-cost* System on Chip (SoC) mikrokontroler dengan microprocessor Tensilica Xtensa LX6 yang terintegrasi dengan WiFi dan Bluetooth (Hübschmann, 2020). ESP32 dirancang untuk lingkungan industri dengan rentang kerja temperature $-40\text{ }^{\circ}\text{C}$ – $125\text{ }^{\circ}\text{C}$ serta memiliki konsumsi energi sangat rendah (*Ultra-low power consumption*) yang menjadikan mikrokontroler ini banyak digunakan untuk pengaplikasian IoT. (Espressif Systems, 2021). Pada penerapannya dibidang IoT, ESP 32 pada *smart industrial*, ESP32 dapat diaplikasikan pada sistem kontrol PLC jarak jauh (Hübschmann, 2020). Bentuk *Hardware* dan *pinout diagram* ESP32 ditunjukkan pada Gambar 2.10.



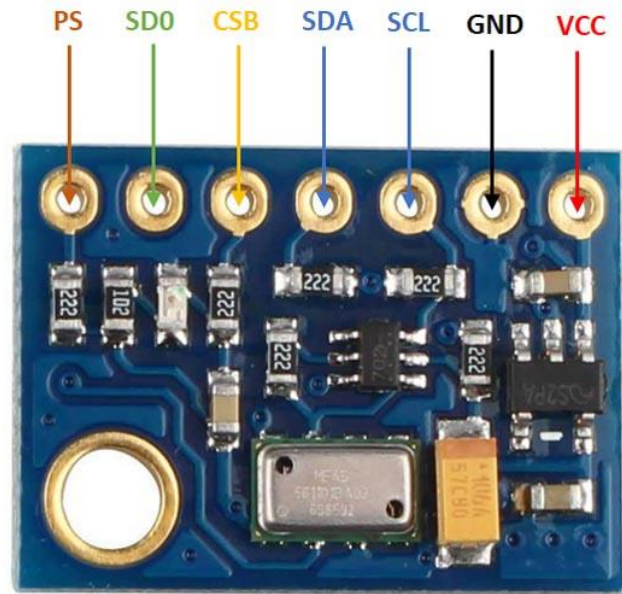
Gambar 2. 10 Mikrokontroler ESP32 dan *Pinout Diagram* (Circuits4you, 2018)

Pada penelitian ini ESP32 difungsikan sebagai kontroler *master* dan *slave*. Kontroler *slave* merupakan kontroler yang berinteraksi dengan *master* untuk melakukan proses akuisisi data dan komputasi algoritma. Data yang diterima oleh kontroler *slave* akan dikirimkan menuju *master* dengan LoRa. Kemudian kontroler *master* mengirimkan datanya pada *control room* yang diakses melalui *cloud database*.

2.9 Barometric Pressure Sensor MS5611

Sensor Tekanan Udara bekerja dengan mengukur tekanan udara berdasarkan perubahan ketinggian atau *volume* atau tekanan udara pada suatu sistem. Sensor tekanan udara yang digunakan ialah MS5611. Sensor ini bekerja dengan rentang 10 hingga 1200 milibar serta memiliki resolusi yang tinggi yaitu 0.012 mbar hingga 0.065 mbar, serta memiliki

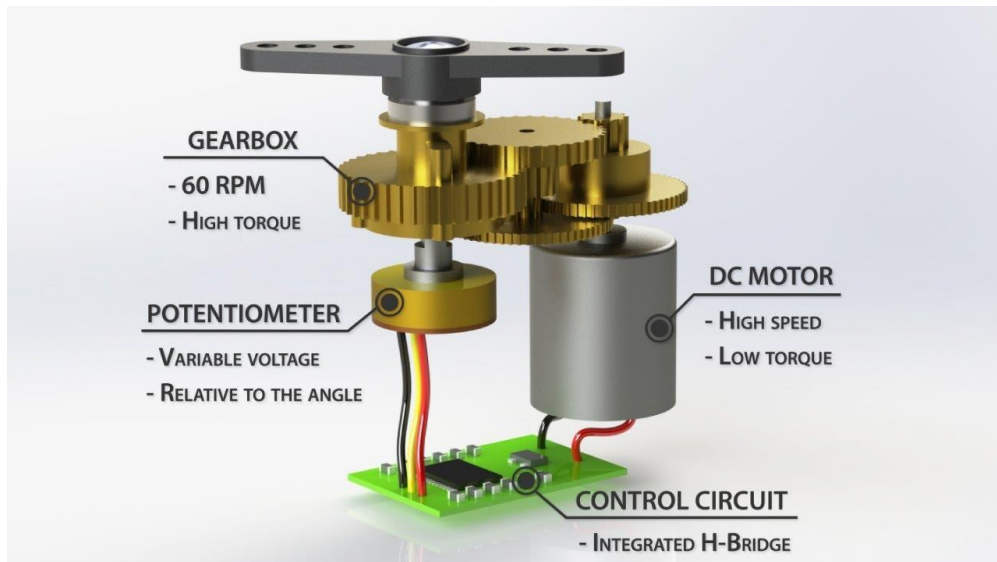
resolusi 24-bit ADC (TE Connectivity Ltd, 2017). Resolusi sensor ini lebih baik dengan yang digunakan sebelumnya. Sensor yang digunakan sebelumnya, BMP180, yang memiliki resolusi 18-bit ADC dengan resolusi pembacaan tekanan hingga 0.02 mbar (Bosch, 2015). Gambar 2.11 menunjukkan bentuk MS5611 dalam *module* dengan *pinout diagram*.



Gambar 2. 11 Sensor MS5611 dan *Pinout Diagram* (Components101, 2021)

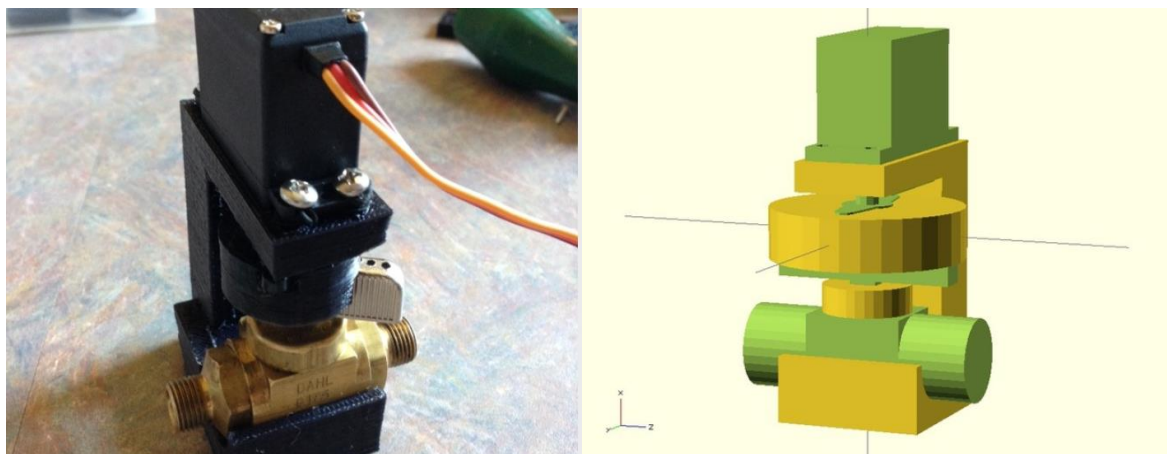
2.10 Servo Motor Control Valve

Control valve difungsikan sebagai *final element* pada kontrol industri proses. Control valve memanipulasi fluida yang mengalir, seperti gas, uap, air, atau senyawa kimia untuk mengkompensasi gangguan beban dan menjaga variabel proses yang diatur sedekat mungkin dengan set point yang diinginkan (Emerson Fisher, 2005). Pada penelitian ini jenis motor yang digunakan dalam valve ialah jenis servo motor. Motor servo memiliki efisiensi yang tinggi serta dapat menyediakan torsi yang cukup pada kecepatan yang tinggi. Pengendalian motor servo dilakukan dengan *Pulse Width Modulation* (PWM). Nilai PWM akan motor servo menentukan posisi poros berdasarkan durasi pulsa yang dikirim melalui kabel sinyal. Secara garis besar, motor servo terbagi atas 4 komponen utama, yaitu motor, *potentiometer*, *control circuit*, dan *gearbox*. *Potentiometer* berperan sebagai sensor yang memberikan umpan balik menuju *control circuit*, sedangkan *control circuit* berperan sebagai sistem kontrol yang mengatur arah putaran motor. Komponen pada motor servo ini diilustrasikan pada Gambar 2.12.



Gambar 2. 12 Bagian Pada Servo Motor (Dejan, 2020)

Pada penelitian ini motor servo digunakan untuk menggerakkan control valve pada outlet minyak dan air berbasis PID dengan *operating voltage* 5V dengan *stall torque* dan *operating speed* secara berturut-turut 9.4 kg/cm dan 0.17 s/60° pada tegangan kerja 4.8 V (Components101, 2019). Servo dipasangkan *joint* yang sesuai dengan kepala *valve* sehingga dapat membuka dan menutup valve secara otomatis menggunakan servo. Mekanisme kerja servo control valve ditunjukkan pada Gambar 2.13.



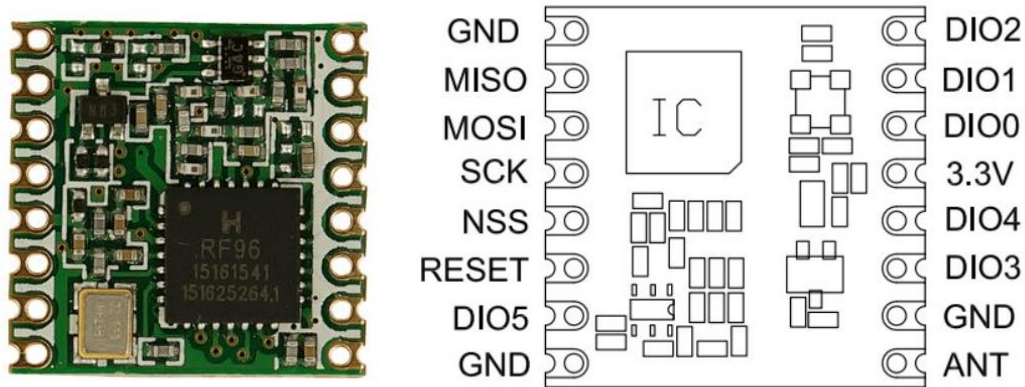
Gambar 2. 13 DC Motor Servo Valve (Parker, 2016)

2.11 Transceiver *Wireless* Long Range RFM95

Modul transceiver untuk berkomunikasi secara *wireless* terhadap plant menggunakan Long Range (LoRa). LoRa merupakan modul komunikasi yang di buat oleh Semtech berbasis *radio frequency* (RF) yang ditujukan untuk komunikasi jarak jauh. LoRa dapat berkomunikasi hingga 5 kilometer pada *urban area* dan mencapai 15 kilometer pada *line of sight*. LoRa menawarkan jarak yang jauh dengan konsumsi yang sangat rendah jika

dibandingkan dengan modul lainnya seperti WiFi, Bluetooth, dan Zigbee serta harga modul yang relatif rendah. (Semtech, 2020).

Modul ini memiliki range frekuensi yang dapat diaplikasikan sesuai dengan peraturan dari Pemerintah Republik Indonesia. Modul yang digunakan dalam penelitian ini terlampir pada Gambar 2.14.



Gambar 2. 14 Modul Komunikasi LoRa RFM95 Beserta *Pinout Diagram* (HOPERF, 2020)

Spesifikasi dari modul komunikasi LoRa RFM95 ditunjukkan pada Tabel 2.3.

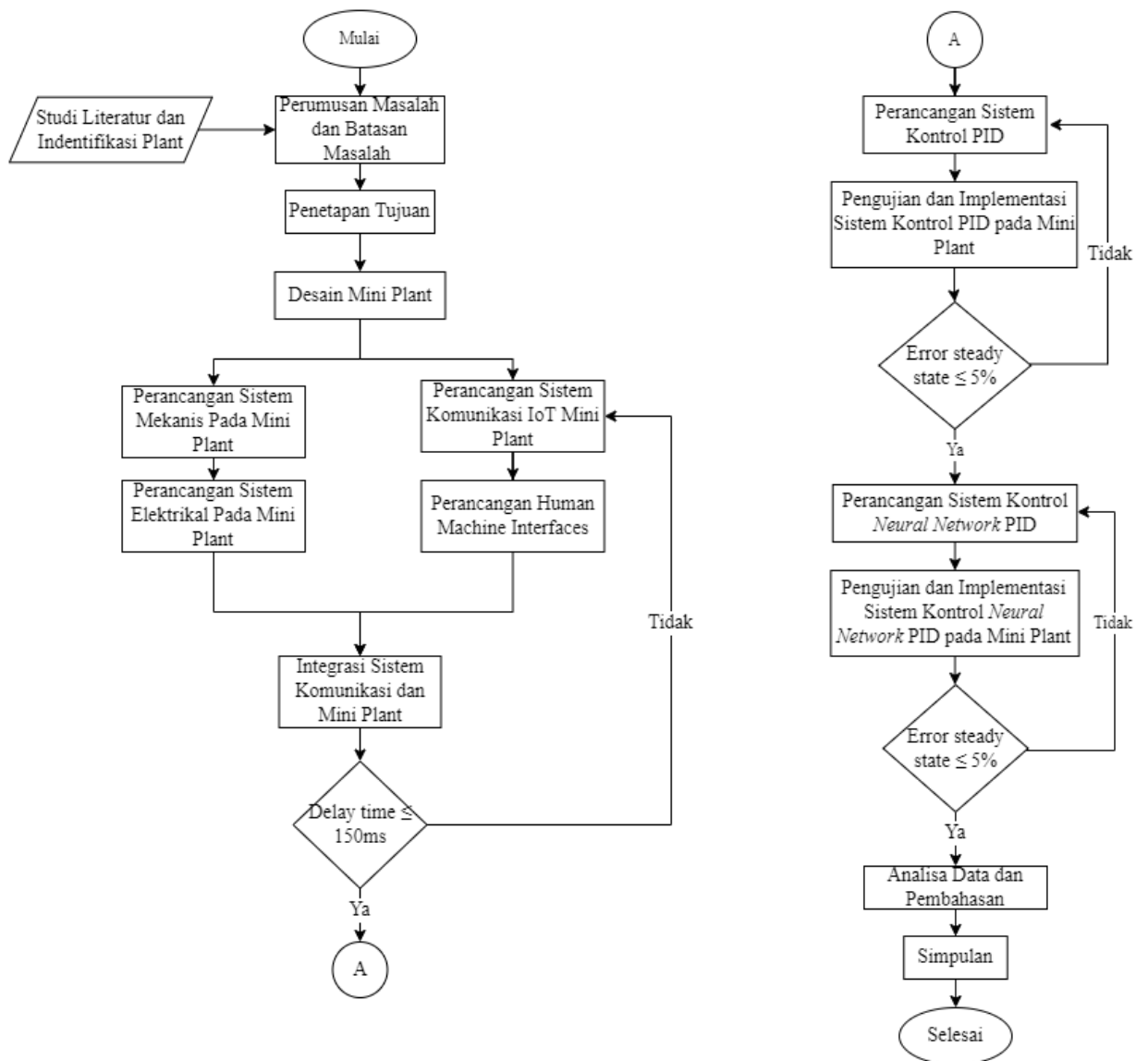
Tabel 2. 3 Spesifikasi LoRa RFM95 (HOPERF, 2020)

SX1276 Low Power Long Range Transceiver	
Frequency Range	137 – 1020 MHz
Supply Voltage	1.8 – 3.7 V
Supply Current in Receiver Mode	11.5 mA
Output Current in Transmission	19.9 mW
Effective Bitrate	37.5 kbps
Sensitivity	-111 to -136 dBm

Penggunaan LoRa setiap negara memiliki regulasi sendiri dalam menggunakan *radio frequency*. Penggunaan modul komunikasi LoRa tercantum pada peraturan Menteri Komunikasi dan Informatika (Kominfo) Republik Indonesia, penggunaan perangkat telekomunikasi *Low Power Wide Area* (LPWA) yang berbasis pita frekuensi harus memenuhi syarat rentang frekuensi 920 – 923 MHz (Kominfo, 2018). Penelitian digunakan modul komunikasi RFM95 yang berfungsi sebagai *transmitter* sekaligus *receiver* antara kontroler *master* dan kontroler *slave*.

BAB III METODOLOGI PENELITIAN

Penelitian ini dilakukan secara bertahap dari desain dan perancangan *mini plant*, perancangan sistem komunikasi dan Human Machine Interfaces (HMI), integrasi sistem komunikasi dan *mini plant*, dan perancangan sistem kendali level pada plan yang ditunjukkan pada diagram alir Gambar 3.1.



Gambar 3. 1 Diagram Alir Metode Penelitian

3.1 Studi literatur

Pada tahap studi literatur dilakukan pencarian terkait informasi dan fakta serta penelitian terkait yang bisa dijadikan pengetahuan untuk menunjang proses penelitian. Studi

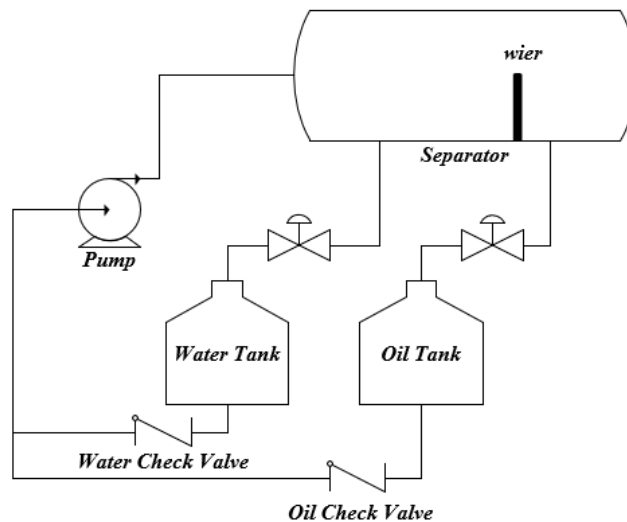
literatur yang dilakukan pada penelitian “Rancang Bangun Sistem Kontrol Level *Mini Plant Separator* Tiga Fasa Menggunakan *Neural Network* PID Berbasis Internet of Things” terkait sistem komunikasi IoT, sistem pengendalian, dan separator yang didapat dari buku, jurnal, dan sumber informasi lainnya yang berkaitan dengan penelitian tersebut baik dari penelitian terdahulu atau kerangka dasar informasi yang digunakan dalam komponen yang digunakan.

3.2 Desain Mini Plant Separator

Proses Desain dan Perancangan Mini Plant meliputi desain skema separator, desain separator, dan perancangan sistem elektrikal separator.

3.2.1 Skema Separator

Berdasarkan prinsip kerja separasi minyak pada industri perminyakan dan gas. Plant pada separator terdiri atas *sensing element* berupa *level transmitter* pada fasa air dan minyak serta aktuator berupa control valve pada fasa air dan minyak. Kemudian dilakukan pembuatan skema separator 3 fasa dilakukan dengan menggunakan fasa air dan minyak sebagai *input*. Perancangan mini plant menggunakan skema berdasarkan Gambar 3.2.



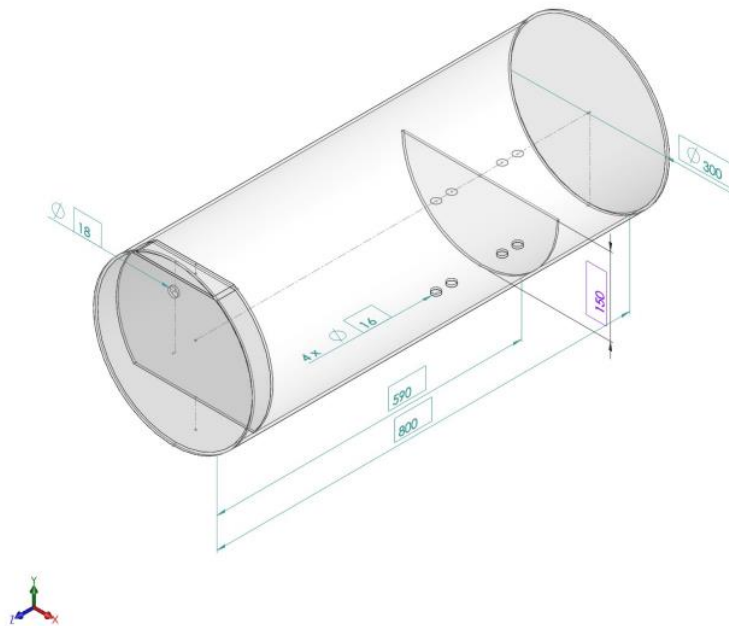
Gambar 3. 2 Skema Separator

Berdasarkan hasil perancangan Skema pada Gambar 3.2, pompa akan mengalirkan campuran yang terdiri atas minyak dan air yang kemudian masuk melewati *inlet* separator. Minyak dan air akan terpisah ke dalam ruang minyak saat sudah melewati *weir*, sedangkan air tertinggal didalam ruang campuran. Ketinggian baik dari ruang campuran dan ruang minyak diukur menggunakan sensor yang masing-masing mengendalikan bukaan *control valve*. Kemudian masing-masing fluida akan masuk ke dalam masing-masing tangki yang tersedia yang kemudian dialirkan kembali ke dalam separator menggunakan pompa.

Keluaran cairan dari kedua tangki dipasang *check valve* untuk menghindari tercampurnya kedua fluida didalam masing-masing tangki.

3.2.2 Perancangan Sistem Mekanik Separator

Berdasarkan penjelasan sebelumnya, skema separator akan di desain sedemikian rupa sehingga menggunakan software CAD untuk mempermudah dalam manufaktur separator. Desain rancangan mini plant didesain menggunakan software Solidworks 2017. Hasil rancangan dasar tabung separator tertera pada Gambar 3. 3.



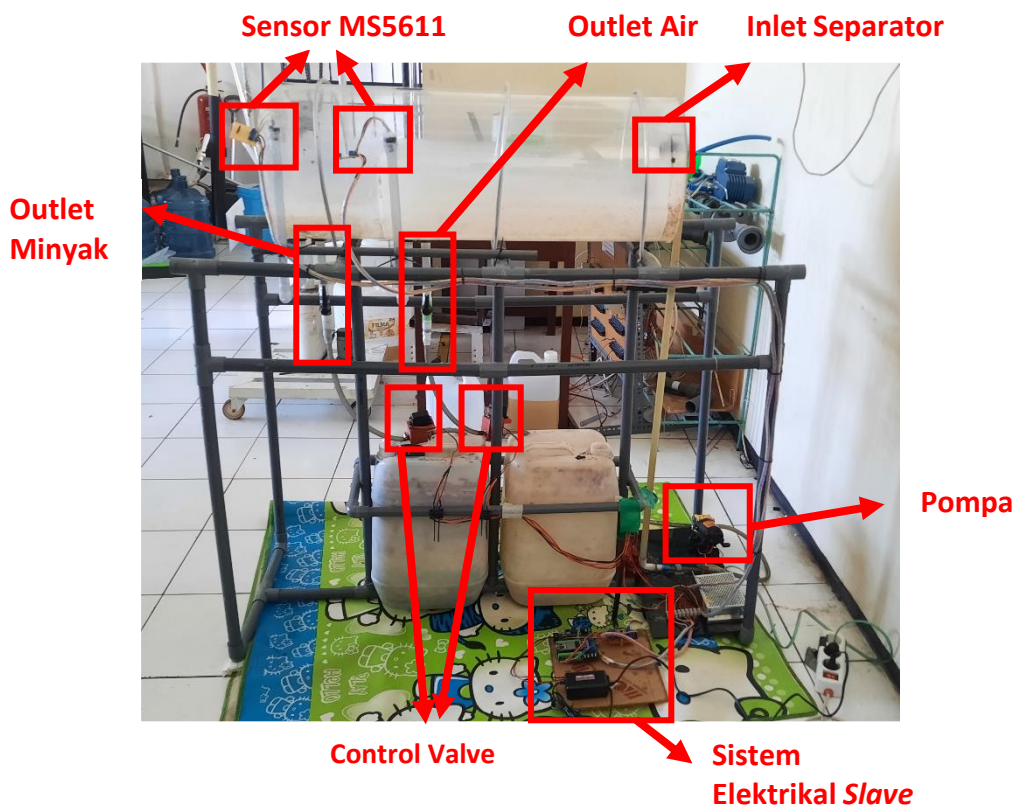
Gambar 3. 3 Desain Tabung Separator

Desain Rancangan Mini Plant Separator menggunakan tabung separator yang sudah ada sebelumnya dengan spesifikasi yang tertera pada Tabel 3.1 sebagai berikut.

Tabel 3. 1 Dimensi Mini Plant Separator

Dimensi	Nilai
Panjang Total	800 mm
Panjang Ruangan	590 mm
Diameter	300 mm
Tinggi <i>weir</i>	150 mm
<i>Nozzle</i> pipa <i>inlet</i>	18 mm
<i>Nozzle</i> pipa <i>outlet</i> air	16 mm
<i>Nozzle</i> pipa <i>outlet</i> minyak	16 mm

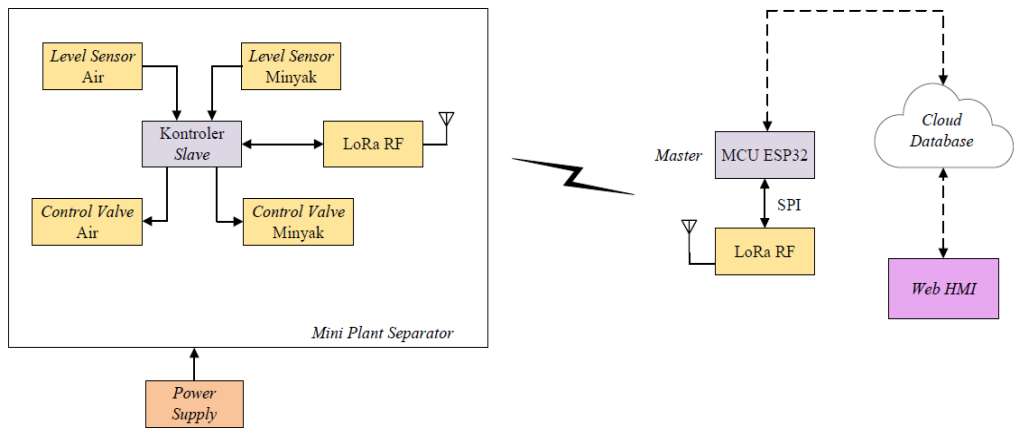
Perancangan secara keseluruhan separator berdasarkan pada Gambar 3.4. Setiap ketinggian ruang diukur melalui sensor MS5611 dengan memanfaatkan tekanan udara dari kevakuman udara pada tabung pembacaan. Semakin tinggi fluida yang ada di dalam tangki, maka akan semakin tinggi pembacaan tekanan udara. Kemudian ketinggian fluida dikontrol menggunakan *servo control valve*. *Manipulated Variable (MV)* dalam sistem ini ialah *flow* yang dihasilkan oleh persen bukaan *control valve* sehingga *Controlled Variable (CV)*, dalam hal ini level cairan, akan terukur oleh sensor yang telah dipasangkan.



Gambar 3. 4 Mini Plant Separator

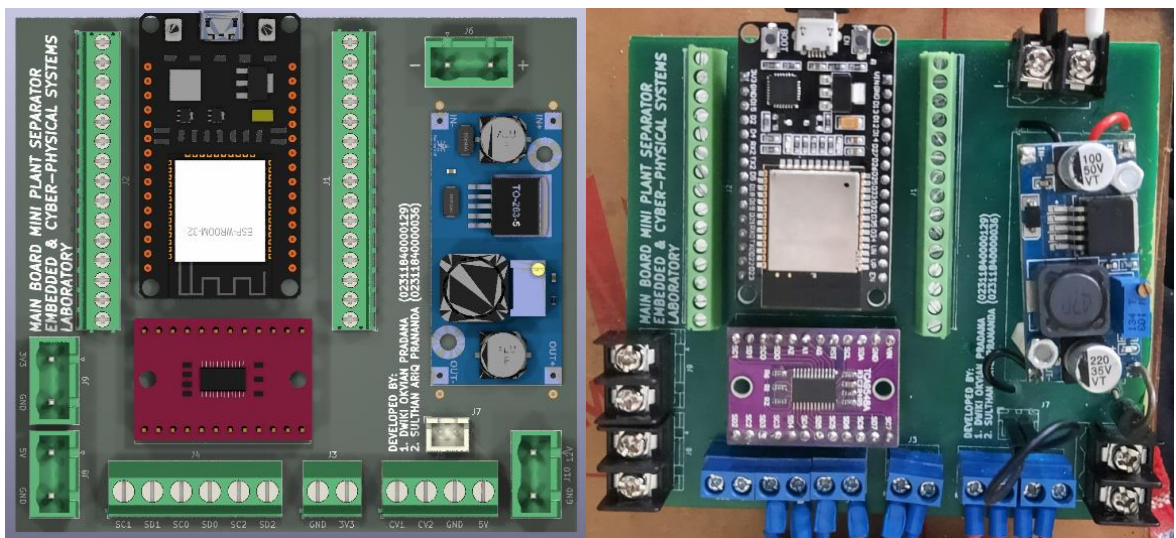
3.2.3 Perancangan Sistem Elektrikal Separator

Perancangan sistem elektrikal separator dibagi menjadi 2, yaitu sistem kontrol *master* dan sistem kontrol *slave*. Sistem kontrol *slave* menaungi sistem sensor node yang terdapat pada plant sebagai sensing element, *actuator* sebagai final element, sistem komunikasi yang terhubung perangkat LoRa, dan *power supply*. Sedangkan sistem kontrol master sistem yang berada pada *control room* yang menerima dan mengirim data menuju *cloud database* yang terhubung dengan sistem komunikasi. Rancangan sistem elektrikal separator ditunjukkan dengan Gambar 3. 5.



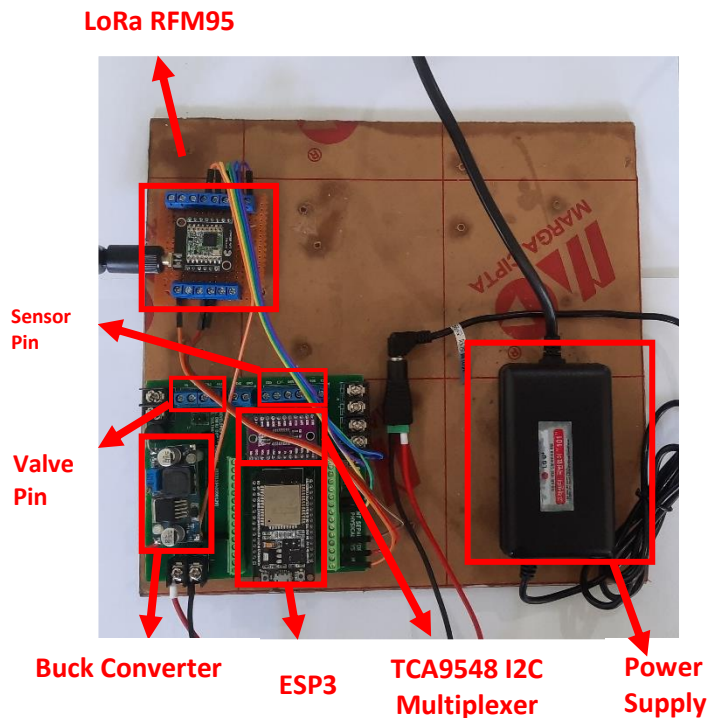
Gambar 3. 5 Rancangan Sistem Elektrikal

Prototype sistem kontrol *slave* dibuat menggunakan *breadboard*. Setelah jalur dilakukan *continuity test* dan *unit test*, maka dirancang pembuatan PCB yang diawali dengan perancangan skematik. Perancangan skematik sudah selesai dan sesuai maka dilanjutkan dengan perancangan *Printed Circuit Board* (PCB) dan dilakukan pencetakan sesuai desain PCB. Hasil perancangan sistem kontrol *slave* ditunjukkan pada Gambar 3.6.



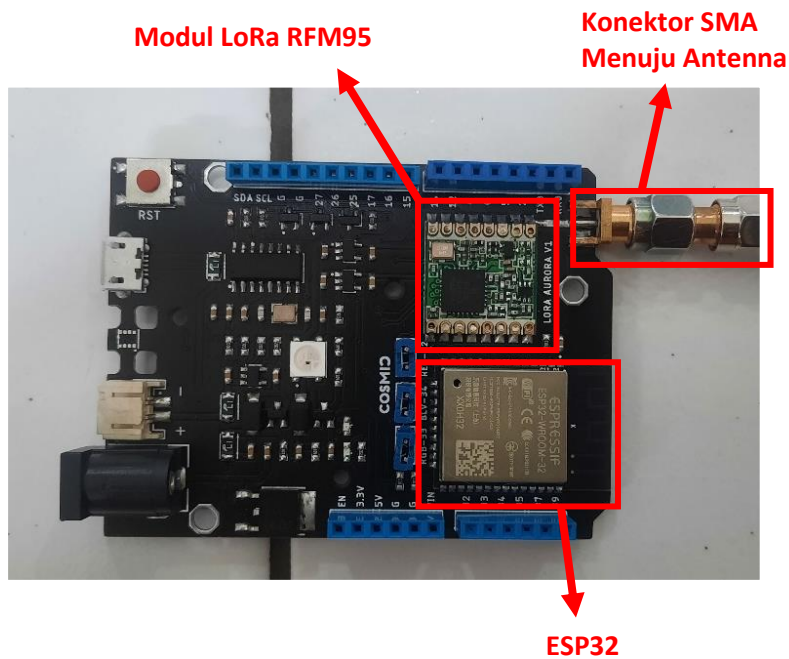
Gambar 3. 6 Hasil Rancangan dan Realisasi Papan Elektrik Sistem Kontroler *Slave*

Kemudian dari rancangan papan sistem kontrol *slave* direalisasikan serta terintegrasi dengan komponen-komponen pendukung lainnya. Hasil keseluruhan rangkaian sistem elektrikal yang beroperasi pada *plant* ditunjukkan dengan Gambar 3.7.



Gambar 3. 7 Rangkaian Keseluruhan Sistem Elektrikal *Plant*

Pada sistem kontroler *master* terhubung langsung dengan modul wifi beserta modul komunikasi perangkat LoRa. Sistem komunikasi LoRa terhubung dengan sistem yang ada di *plant*, serta sistem modul wifi menghubungkan koneksi internet untuk mengirim data menuju *cloud database*. Rangkaian kontroler *master* terdapat pada Gambar 3.8.



Gambar 3. 8 Rangkaian Kontroler *Master*

Setelah perancangan sistem kontroler *master* dan *slave* juga dilakukan pengujian sensor untuk menguji apakah sensor dapat bekerja dengan baik yang kemudian dikalibrasi. Penelitian ini menggunakan modul pengukuran level menggunakan sensor tekanan MS5611. Pembacaan tekanan pada saat ketinggian *min* (0 cm) dan *max* (18 cm) diukur, kemudian dilakukan kalibrasi dengan fungsi *mapping* untuk mendapatkan rentang ketinggian separator. Perhitungan level menggunakan persamaan (3.1) dan hasil pembacaan ditampilkan pada Tabel 3.2.

$$h = h_{min} + \left(\frac{P - P_{min}}{P_{max} - P_{min}} \right) (h_{max} - h_{min}) \quad (3.1)$$

dengan

h = level pembacaan (cm)

h_{max} = level maksimum (cm)

h_{min} = level minimum (cm)

P = pressure pembacaan (Pa)

P_{max} = pressure maksimum (Pa)

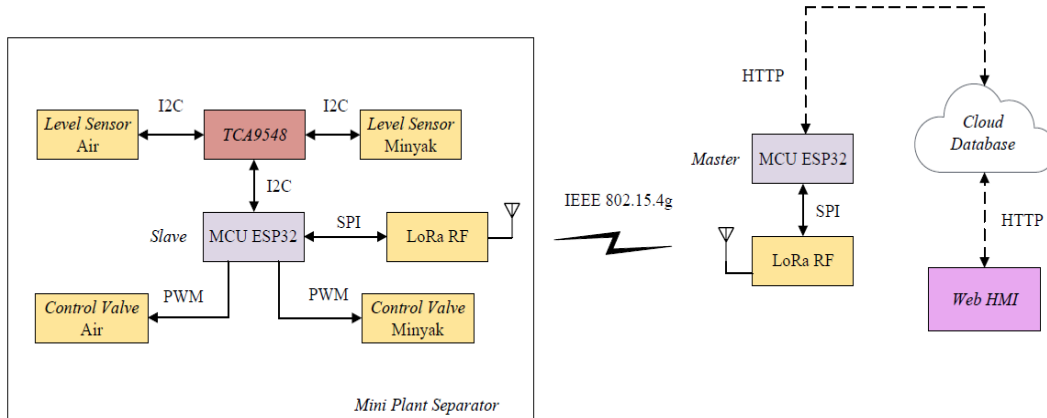
P_{min} = pressure minimum (Pa)

Tabel 3. 2 Penyesuaian Modul Sensor Untuk Pengukuran Level

Parameter	Ruang Campuran	Ruang Minyak
Pressure Min (Pa)	101485	100425
Pressure Max (Pa)	103000	101900
Level Min (cm)	0	0
Level Max (cm)	18	18

3.3 Perancangan *Human Machine Interfaces* dan Sistem Komunikasi *plant*

Perancangan *Human Machine Interfaces* (HMI) diperlukan untuk memudahkan operator dalam berinteraksi dengan proses yang terjadi. Perancangan HMI menggunakan platform berbasis Web HMI dengan *framework* React JS. Perancangan sistem IoT terintegrasi dengan perangkat LoRa yang terhubung secara Point-to-Point (P2P) Communication.



Gambar 3. 9 Skema Sistem Komunikasi *Plant*

Secara keseluruhan sistem komunikasi pada plant terbagi atas komunikasi berbasis kabel dan nirkabel. Pada Gambar 3.9 terlihat bahwa kontroler *master* terhubung secara kabel dengan modul TCA9548 yang berfungsi membaca kedua sensor (campuran dan minyak) dengan protokol komunikasi I2C serta terhubung secara SPI dengan modul komunikasi LoRa. Aktuator terhubung dengan kontroler *slave* dengan mengirimkan sinyal PWM menuju motor servo valve. Pada kontroler *master* juga terhubung dengan LoRa dengan protokol komunikasi SPI, kemudian kontroler *master* mengirimkan data yang diterima melalui komunikasi LoRa menuju *cloud database* dengan koneksi internet via WiFi. Sistem komunikasi antara kontroler *master* dan *slave* terhubung secara nirkabel dengan protokol komunikasi IEEE 802.15.4g yang saling terhubung menggunakan modul komunikasi LoRa.

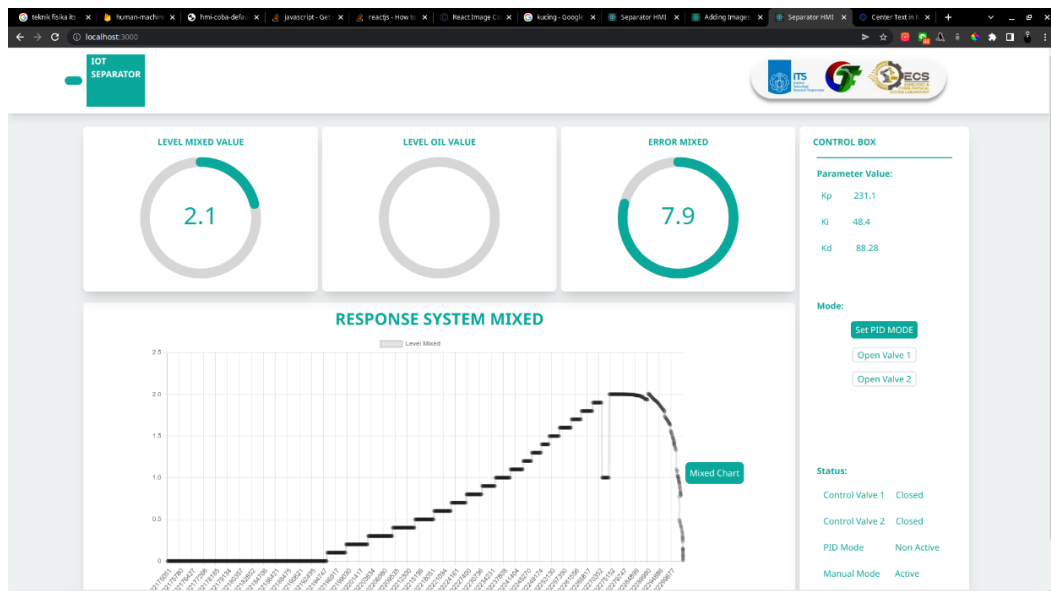
3.3.1 Perancangan Komunikasi *Wireless Sensor and Actuator Networks*

Perancangan sistem komunikasi dilakukan secara P2P dengan frekuensi rentang frekuensi berada di 920-923 MHz sesuai dengan rentang frekuensi yang diizinkan (Kominfo, 2018). Perangkat LoRa berkomunikasi menggunakan protokol komunikasi IEEE 802.15.4g. Setiap sistem pada P2P ini dapat berperan menjadi *transmitter* dan *receiver (transceiver)* sehingga dapat berkomunikasi dua arah. Pada setiap transceiver frekuensi yang digunakan ialah 920MHz, Spreading Factor (SF) 7, serta register sinkronisasi setiap perangkat.

Pada kontroler *master*, data diperoleh dari kontroler *slave* akan diteruskan menuju kontroler *master*. Sistem board kontroler *master* memiliki modul WiFi sehingga data yang diteruskan dari kontroler *slave* akan dikirimkan menuju *cloud database* dengan protokol HTTP.

3.3.2 Perancangan *Human Machine Interfaces*

Perancangan *Human Machine Interfaces* (HMI) dilakukan berbasis Web HMI dengan *framework React Js* agar dapat melakukan rendering secara *real-time*. Perancangan HMI dilakukan dengan penentuan kebutuhan data yang ditransmisikan menuju *cloud*, kemudian setelah ditentukan kebutuhan data yang dimonitor maka dilakukan perancangan desain dari interfaces itu sendiri. Setelah itu dilakukan integrasi dengan *cloud database* apakah request API yang telah dilakukan sudah sesuai atau belum, kemudian dilakukan percobaan tanpa *interfaces* pada *framework* untuk memastikan data sampai diterima atau belum serta melihat kecocokan data. Kemudian setelah semua telah di *set up*, maka dilanjutkan dengan integrasi keseluruhan sistem dengan memuat data dari kontroler *slave* hingga berujung di HMI. Kecocokan data yang masuk diperhatikan antara kontroler *slave* hingga HMI. HMI juga berkomunikasi dengan protokol HTTP untuk mengirim dan menerima data menuju *cloud database*. HMI dan kontroler *master* dianggap klien dalam sistem ini, sehingga perubahan yang terjadi disebabkan oleh HMI akan diterima secara langsung oleh kontroler *master*, berlaku sebaliknya. Akses data dari *cloud database* membutuhkan API *key* sebagai kunci untuk autentikasi akses dari *cloud database*, sedangkan pengambilan dan pengiriman data menggunakan format JSON sehingga untuk mengirim dan menuliskan data ke dalam *cloud database* dibutuhkan *address* object dari struktur format JSON yang telah dibuat serta API *key* yang sesuai.



Gambar 3. 10 Tampilan HMI

Pada Gambar 3.10 merupakan tampilan HMI yang terdiri atas fitur utama untuk visualiasi data level campuran dan minyak beserta errornya dari *setpoint* yang telah ditetapkan serta grafik respon dinamik dari level cairan. yang terdapat *circular progress bar*.

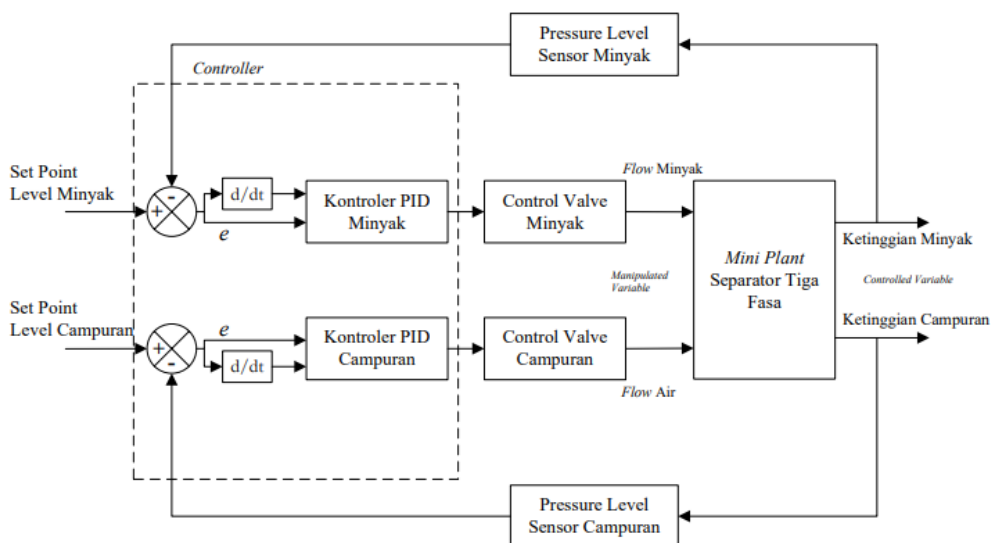
Pada bagian kanan *interface* terdapat *sidebar* berupa *control box* yang akan memperlihatkan perubahan K_p , K_i , dan K_d pada *plant*. Pada *sidebar* terdapat sistem kendali *plant* dengan terdapat *button* untuk set mode pada PID dan membuka atau menutup kedua *control valve*. HMI ini memungkinkan juga untuk melakukan *logging* data dengan menyimpan data ke dalam bentuk format csv saat dilakukan action dengan menekan tombol log pada *sidebar*. Sistem pada HMI ini menerima dan mengirim data sehingga dapat melakukan monitoring dan kontrol sistem.

3.4 Integrasi *Plant*, Sistem Komunikasi, dan *Human Machine Interface*

Pada tahap ini dilakukan *system testing* secara menyeluruh untuk mengetahui apakah dari *plant* dapat mengirim menuju *interface* dengan baik tanpa ada kehilangan data. Juga dilakukan sebaliknya untuk mengetahui apakah dari sistem *interface* dapat mengirim data menuju sistem pada *plant* tanda ada kehilangan data. Jika seluruh rangkaian pada tahapan ini berhasil, maka sudah sesuai kriteria dan dapat dilanjutkan ke dalam perancangan sistem pengendalian.

3.5 Perancangan dan Implementasi Sistem Pengendalian PID

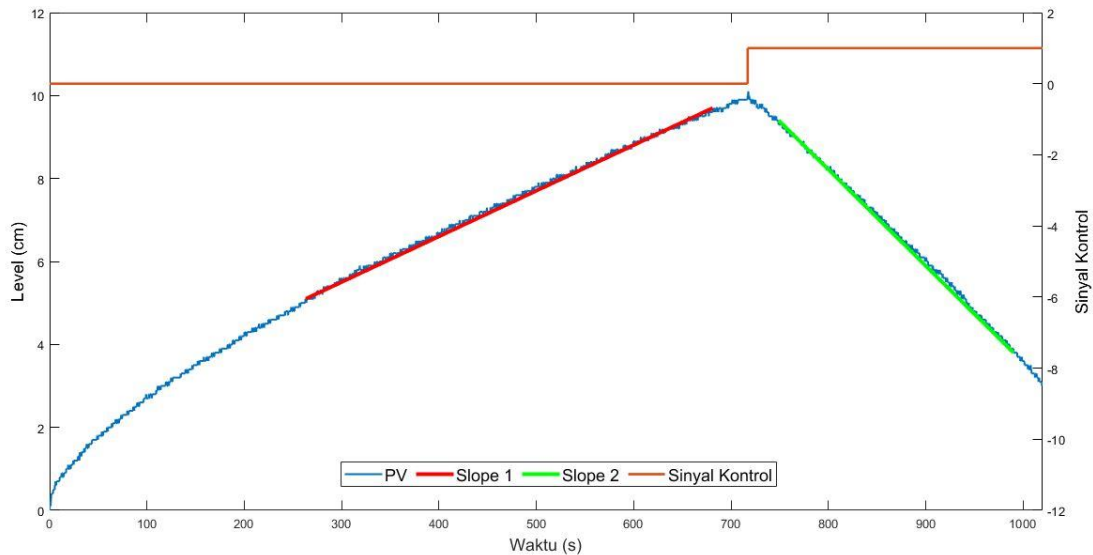
Perancangan sistem pengendalian PID didasarkan pada referensi yang telah dijelaskan sebelumnya, dimana sistem pengendalian level termasuk kedalam *integrating process* yang dimana *process variable* akan membentuk *slope* akibat *step input*. Perancangan sistem pengendali ini dilakukan dengan melihat merancang diagram blok sistem pengendali pada *mini plant* separator. Diagram blok dalam perancangan pengendalian PID seperti pada Gambar 3.11.



Gambar 3. 11 Diagram Blok Sistem Pengendalian PID

3.5.1 Uji Open Loop Sistem

Pengujian open loop sistem dilakukan berdasarkan hubungan antar gradien dengan mengalirkan fluida menuju separator dengan laju konstan yang telah ditentukan, kemudian saat mencapai titik tertentu valve dibuka sehingga terbentuk 2 slope *process variable*. Hasil respon dari pengujian open loop dari level campuran ditunjukkan pada Gambar 3.12 dan 3.13.



Gambar 3. 12 Respon Open Loop Campuran

Berdasarkan Gambar 3. 12, didapatkan beberapa parameter penting yang akan digunakan yang terlampir pada Tabel 3.3.

Tabel 3. 3 Parameter Respon Open Loop Campuran

Parameter	Nilai	Parameter	Nilai
PV1 (cm)	5.1	PV3 (cm)	3.8
PV2 (cm)	9.7	PV4 (cm)	9.4
T1 (menit)	6.966	T2 (menit)	4.022
t_d (s)	0.955	PV Maksimal	10.1
dCO	100 %	PV Minimum	0

Data pada Tabel 3.3 kemudian dilakukan kalkulasi perhitungan untuk mendapatkan parameter PID yang diawali dengan menggunakan persamaan (2.2), (2.3), dan (2.4) sebagai berikut.

$$|DS| = \frac{(3.8 - 9.4)}{4.022} - \frac{(9.7 - 5.1)}{6.966} = 2.053$$

$$DS (\%) = 100 \times \frac{2.053}{(10,1 - 0)} = 20.323$$

$$r_i = \frac{20.323}{100} = 0.2032$$

Untuk mendapatkan parameter control PID, dilakukan perhitungan K_c , T_i , dan T_d berdasarkan aturan metode control loop berdasarkan persamaan pada Tabel 2.2.

$$K_c = \frac{0.75}{0.2032 \times 0.955} = 3.8644$$

$$T_i = 5 \times 0.955 = 4.775$$

$$T_d = 0.4 \times 0.955 = 0.382$$

Nilai diatas diubah kedalam parameter K_p , K_i , dan K_d dengan menggunakan persamaan (3.1), (3.2), dan (3.3)

$$K_p = K_c \tag{3.1}$$

$$K_i = \frac{K_c}{T_i} \tag{3.2}$$

$$K_d = K_c \times T_d \tag{3.3}$$

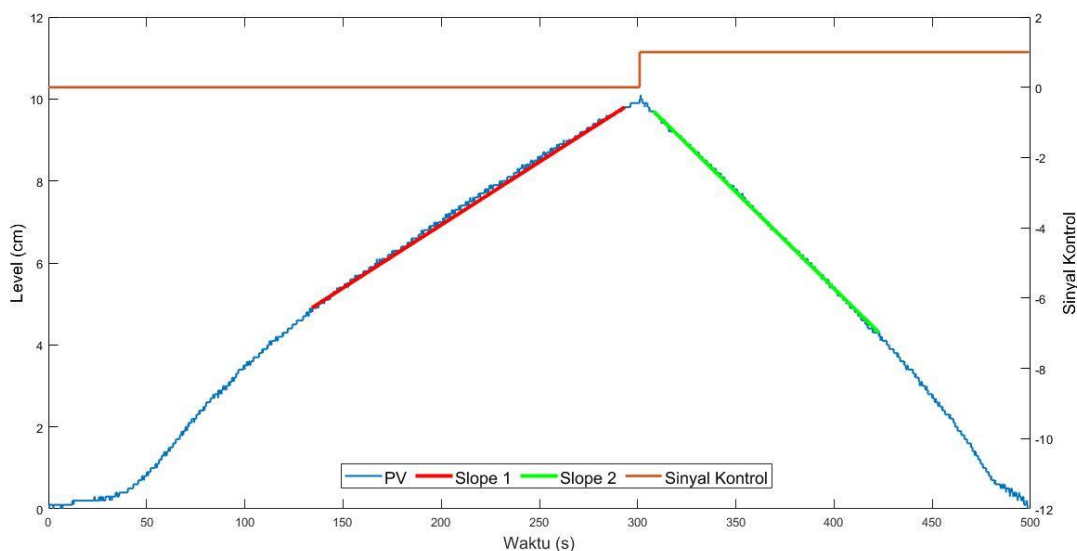
Dengan persamaan diatas, maka didapatkan nilai parameter PID sebagai berikut.

$$K_p = 3.8644$$

$$K_i = \frac{3.8644}{4.775} = 0.8093$$

$$K_d = 3.8644 \times 0.382 = 1.4762$$

Dengan cara yang sama, maka dilakukan pula perhitungan pada respon open loop pada fasa minyak sebagai berikut.



Gambar 3. 13 Respon Open Loop Minyak

Berdasarkan Gambar 3. 13, didapatkan beberapa parameter penting yang akan digunakan yang terlampir pada Tabel 3.4.

Tabel 3. 4 Parameter Respon Open Loop Minyak

Parameter	Nilai	Parameter	Nilai
PV1 (cm)	4.9	PV3 (cm)	2.5
PV2 (cm)	9.7	PV4 (cm)	9.7
T1 (menit)	2.634	T2 (menit)	2.448
td (s)	1.359	PV Maksimal	10.1
dCO	100 %	PV Minimum	0

Data pada Tabel 3.4 kemudian dilakukan kalkulasi perhitungan untuk mendapatkan parameter PID yang diawali dengan menggunakan persamaan (2.2), (2.3), dan (2.4) sebagai berikut.

$$|DS| = \frac{(2.5 - 9.7)}{2.448} - \frac{(9.7 - 4.9)}{2.634} = 4.764$$

$$DS (\%) = 100 \times \frac{4.7637}{(10,1 - 0)} = 47.166$$

$$r_i = \frac{47.166}{100} = 0.4717$$

Maka didapatkan nilai Kc, Ti, dan Td dari persamaan pada Tabel 2.2 sebagai berikut.

$$K_c = \frac{0.75}{0.4717 \times 1.359} = 1.170$$

$$T_i = 5 \times 1.359 = 6.795$$

$$T_d = 0.4 \times 1.359 = 0.5436$$

Sehingga nilai Kp, Ki, dan Kd pada hasil open loop fasa minyak menjadi

$$K_p = 1.170$$

$$K_i = \frac{1.170}{6.795} = 0.172$$

$$K_d = 1.170 \times 0.5436 = 0.636$$

Dengan perhitungan yang telah dilakukan, maka didapatkan nilai parameter PID hasil uji open loop seperti pada Tabel 3.5.

Tabel 3. 5 Hasil Parameter PID dari Pengujian Open Loop

Parameter PID	PID Fasa Campuran	PID Fasa Minyak
Kp	3.8644	1.170
Ki	0.8093	0.172
Kd	1.4762	0.636

Dengan parameter diatas, dilakukan selanjutnya pengujian performansi apakah respon dinamik *error steady state* dibawah 5%. Jika belum memenuhi, maka dilakukan *fine tuning* hingga mencapai error kriteria respon dinamik *error steady state* dibawah 5%.

3.5.2 Perancangan dan Implementasi Sistem Pengendalian PID

Sistem pengendalian PID dirancang berdasarkan persamaan (2.2) yang ditulis kembali dan ditanamkan ke mikrokontroler dalam bentuk PID diskrit. Persamaan PID digital yang digunakan merupakan persamaan (3.4).

$$u(n) = u(n - 1) + [K_p(e_n - e_{n-1}) + K_i e_n + K_d(e_n - 2e_{n-1} - e_{n-2})] \quad (3.4)$$

dengan:

$u(n)$ = Sinyal Kontrol

$u(n - 1)$ = Sinyal Kontrol Sebelumnya

K_p = *Proportional Gain*

K_i = *Integral Gain*

K_d = *Derivative Gain*

$e(n)$ = *Error* saat ini

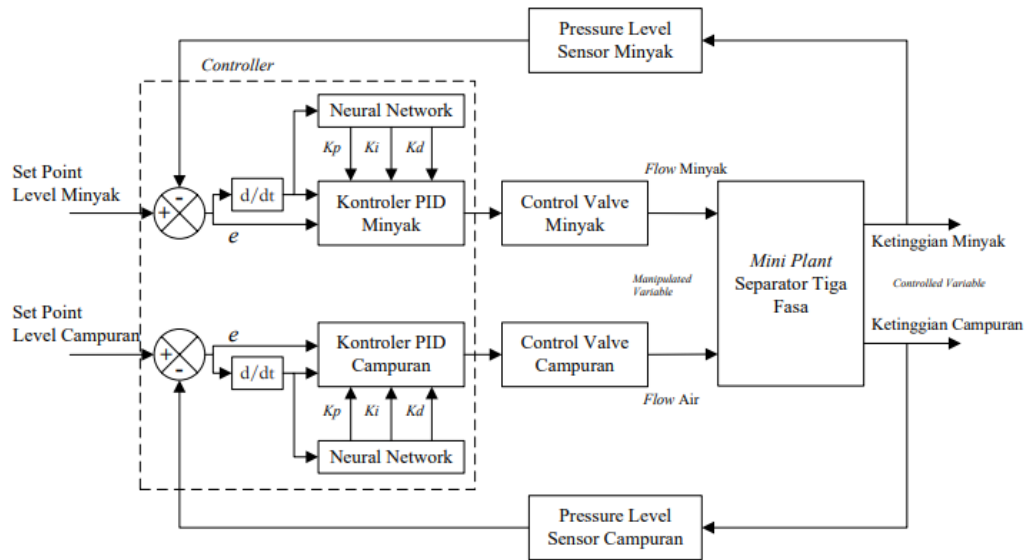
$e(n - 1)$ = *Error* sebelumnya

PID dirancang dalam mode *direct acting* yang secara kontinu melakukan komputasi PID berdasarkan *input* tuning parameter PID. Hasil dari parameter dari pengujian open loop ini kemudian dimasukkan sebagai *initial parameter* dalam *input* parameter PID.

3.6 Perancangan dan Implementasi *Neural Network*

Perancangan algoritma *Neural Network* ditujukan untuk menentukan kondisi *plant* sesuai target dengan melakukan *tuning* parameter PID secara adaptif. Hasil komputasi *Neural Network* akan mengirimkan *input* parameter PID menuju pengendali PID. Komputasi *Neural Network* mengambil *input* data berupa *error* dan *delta error* yang dituangkan ke dalam *state*. Komputasi algoritma *Neural Network* dilakukan didalam mikrokontroler.

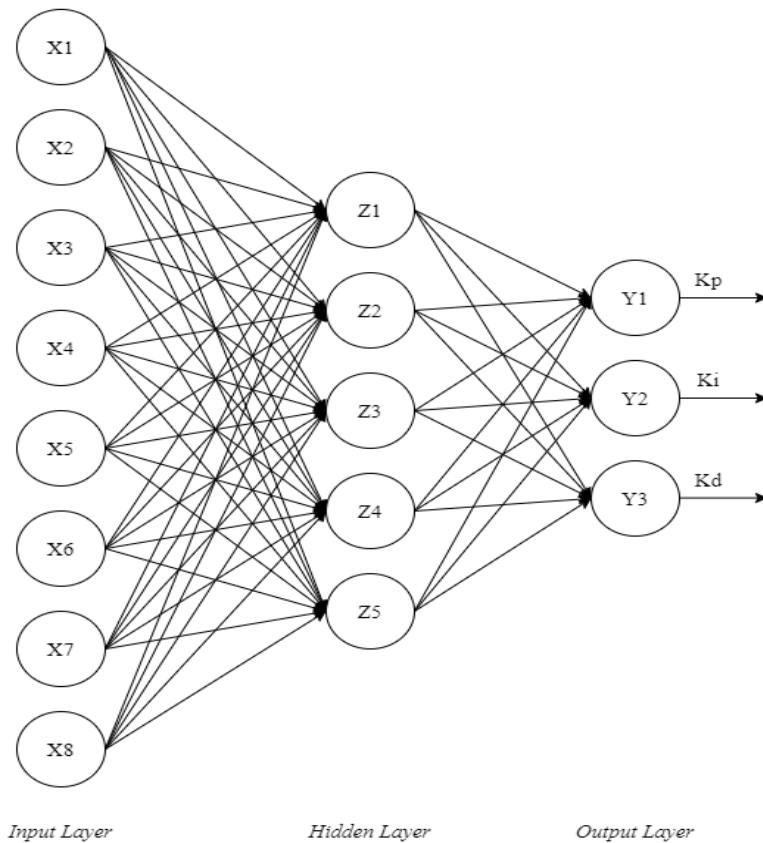
Perancangan sistem pengendalian *Neural Network*-PID berdasarkan diagram blok yang terdapat pada Gambar 3.14.



Gambar 3. 14 Diagram Blok Sistem Pengendalian *Neural Network* PID

3.6.1 Perancangan Arsitektur *Neural Network*

Dalam perancangan arsitektur *Neural Network*, ditentukan terlebih dahulu metode yang akan digunakan. Perancangan arsitektur menggunakan *Neural Network feed forward* dengan algoritma pelatihan *backpropagation*. Kemudian ditentukan struktur dari *Neural Network* itu sendiri, dimana hal ini merupakan penggunaan jumlah *layer* yang digunakan dan jumlah *node* pada setiap *layer*. Perancangan arsitektur *Neural Network* pada penelitian ini berbasis *Multi Layer Perceptron* (MLP) dimana terdiri atas 8 neuron pada *input layer*, 5 neuron pada *hidden layer*, dan 3 neuron pada *output layer* yang merupakan 3 dari representasi dari parameter PID (K_p , K_i , dan K_d). Bentuk struktur rancangan *neural network* yang dibuat berdasarkan Gambar 3.15.



Gambar 3. 15 Struktur Perancangan *Neural Network* pada Pengendali *Neural Network* PID

Pada hidden layer dan output layer rancangan ini menggunakan fungsi aktivasi *sigmoid biner*. Persamaan sigmoid memiliki output 0 hingga 1 dalam bentuk non-linier seperti persamaan (2.7). Penggunaan sigmoid dikarenakan kebutuhan output bersifat non-linier dan parameter PID bersifat non-negatif.

Pada perancangan ini *input* berupa kombinasi nilai biner yang merepresentasikan variabel yang terjadi pada variabel *input error* dan *delta error*. Kemudian *input* masuk ke dalam *hidden layer* berdasarkan persamaan (2.6). Pada *output layer* menggunakan alur komputasi yang sama seperti *hidden layer*. Hasil output layer kemudian akan merepresentasikan nilai K_p , K_i , dan K_d yang didapat dari persamaan (2.10). Nilai keluaran *output layer* belum bisa digunakan secara langsung, ini dikarenakan rentang output berada direntang 0 hingga 1. Keluaran dari *output layer* pada persamaan (3.7), (3.8), dan (3.9)

$$y_1 = \frac{K_p}{100} \quad (3.7)$$

$$y_2 = \frac{K_i}{100} \quad (3.8)$$

$$y_3 = \frac{Kd}{100} \quad (3.9)$$

Output ini dilakukan normalisasi kembali dengan dikalikan 100 agar sesuai dengan nilai parameter yang diharapkan.

Pada algoritma pelatihan *backpropagation*, algoritma ini bekerja mengevaluasi error yang terjadi dan melakukan komputasi dengan alur yang sebaliknya dari *feed forward*.

Pelatihan ini dilakukan secara terus menerus dan berhenti apabila sudah mencapai iterasi maksimum serta nilai dari target error yang diharapkan. Saat sudah memenuhi target *error* minimum, maka nilai bobot pada *hidden layer* dan *output layer* disimpan untuk melakukan komputasi *Neural Network* secara *feed forward*. Perhitungan error berdasarkan persamaan (3.10) *mean square error* (MSE) sebagai berikut.

$$E_k = \frac{1}{2} (t_k - y_k)^2 \quad (3.10)$$

dengan:

E_k = Performansi error pada *output layer*

t_k = Target yang diharapkan pada node k *output layer*

y_k = Output pada node k *output layer*

Secara keseluruhan maka parameter perancangan model *Neural Network* yang telah dibuat dirangkum dalam Tabel 3.6.

Tabel 3. 6 Parameter Model JST

No	Parameter	Keterangan
1	Arsitektur	MLP <i>Feed Forward</i>
2	Algoritma Pelatihan	<i>Backpropagation</i>
3	Jumlah <i>Hidden Layer</i>	1
4	Jumlah Node <i>Hidden Layer</i>	5
5	Target MSE	0.001
6	<i>Learning Rate</i>	0.2
8	Fungsi Aktivasi	Sigmoid

3.6.2 Variabel *Input Output Neural Network*

Pemilihan variabel *input* dan *output* berdasarkan data yang dimiliki. Data yang dimiliki berdasarkan percobaan yang dilakukan sebelumnya pada hasil *fine tuning* dan data pada Lampiran C dan D serta pedoman *tuning* parameter PID pada Tabel 2.2. Perancangan *Neural Network* ini memiliki 2 *input* yang terdiri dari *error* serta perubahan *error* (*delta*

error) serta 3 output yaitu parameter PID Kp, Ki, dan Kd. Variabel *input* dari *error* ialah state dari *error* saat nol, diantara 0.1 hingga 0.5, diantara -0.5 hingga -0.1, lebih dari 0.5, kurang dari -0.5. Sedangkan variabel *input delta error* ialah nol, lebih dari 0.1, dan kurang dari -0.1. Untuk variabel output Kp, Ki, dan Kd terdiri dari kombinasi nilai *low*, *medium*, dan *high*. Tabel 3.7 dan 3.8 merupakan tabel masukan dari *error* dan *delta error*.

Tabel 3. 7 Masukan *Error*

Input Error					
<i>State</i>	X1	X2	X3	X4	X5
0	1	0	0	0	0
0.1 – 0.5	0	1	0	0	0
> 0.5	0	0	1	0	0
-0.5 – -0.1	0	0	0	1	0
< -0.5	0	0	0	0	1

Tabel 3. 8 Masukan *Delta Error*

Input Delta Error			
<i>State</i>	X6	X7	X8
0	1	0	0
>= 0.1	0	1	0
<= -0.1	0	0	1

Ditentukan juga variabel keluaran yang dapat merepresentasikan parameter PID. Perancangan output parameter PID terdiri atas kombinasi dari Kp, Ki, dan Kd yang berdasarkan pengujian performansi respon dinamik pada *plant*. Output parameter direpresentasikan melalui nilai dilampirkan di dalam Tabel 3.9.

Tabel 3. 9 Keluaran Parameter PID

Output Parameter PID			
<i>Error\deltaError</i>	0	>= 0.1	<= -0.1
0	T1	T2	T2
0.1 – 0.5	T3	T4	T4
> 0.5	T5	T6	T6
-0.5 – -0.1	T3	T4	T4
< -0.5	T5	T6	T6

Mode kombinasi pada setiap target didapatkan berdasarkan pengetahuan penulis dan hasil pengujian dari hasil *fine tuning* serta pengujian respon dinamik lainnya dengan kombinasi 3 parameter PID. Kombinasi parameter PID yang digunakan pada level campuran terdapat pada Tabel 3.10 dan level minyak terdapat pada Tabel 3.11.

Tabel 3. 10 Tabel Kombinasi Parameter PID Level Campuran

Mode	Kp	Ki	Kd
T1	45	5	15
T2	45	5	25
T3	65	20	3
T4	65	50	10
T5	95	25	15
T6	95	25	25

Tabel 3. 11 Tabel Kombinasi Parameter PID Level Minyak

Mode	Kp	Ki	Kd
T1	18	4.5	20
T2	18	4.5	20
T3	35	6	30
T4	35	6	50
T5	40	15	20
T6	40	15	50

Rancangan variabel *input* output pada *neural network* secara keseluruhan terdapat pada Lampiran B.

3.6.3 Inisialisasi Bobot *Neural Network*

Inisialisasi pada *neural network* dilakukan pada bobot yang ada di *hidden layer* dan *output layer*. Inisialisasi bobot awal dilakukan dengan nilai acak dengan rentang 0 hingga 1 menggunakan fungsi random. Output inisialisasi diset dengan maksimal bobot 0.5 dengan menggunakan persamaan (3.11)

$$w_{init} = 2 \times \left(\frac{random(100)}{100} - 0.5 \right) w_{initMaks} \quad (3.11)$$

dengan:

w_{init} = Inisialisasi bobot pada *hidden layer* dan *output layer*

$w_{initMaks}$ = Bobot maksimum yang diharapkan pada *hidden layer* dan *output layer*

Perubahan bobot pada hidden layer diawali dengan persamaan (2.11) untuk menghitung informasi *error* dari *output layer* menuju *hidden layer* yang ditulis kembali dalam persamaan (3.12) dengan derivatif dari fungsi aktivasi *sigmoid* menjadi

$$\delta_k = (t_k - y_k) \times y_k \times (1 - y_k) \quad (3.12)$$

yang kemudian perubahan bobot dilakukan dengan menggunakan persamaan (2.18). Kemudian pada kalkulasi informasi *error hidden layer* menuju *input layer* dilakukan dengan menggunakan persamaan (2.15) yang ditulis kembali menjadi persamaan (3.13)

$$\delta_j = \sum_{k=1}^m \delta_k w_{jk} + z_j \times (1 - z_j) \quad (3.13)$$

yang kemudian perubahan bobot dilakukan dengan menggunakan persamaan (2.19).

Setelah dilakukan pelatihan, nilai setiap bobot pada *hidden layer* dan *output layer* disimpan untuk digunakan selanjutnya untuk *feed forward neural network*. Hasil iterasi pelatihan dan MSE terakhir terlampir pada Tabel 3.12.

Tabel 3. 12 Jumlah Iterasi dan MSE Setelah Pelatihan Selesai

Jenis	Jumlah Iterasi	MSE Terakhir
Level Campuran	183	0.00099
Level minyak	374	0.00095

Hasil inisialisasi bobot *hidden layer* dan *output layer* untuk melakukan komputasi secara *feed forward*. Bobot yang sudah sesuai dengan nilai MSE yang diharapkan ini digunakan untuk pengambilan data respon dinamik *mini plant* dengan pengendali *Neural Network* PID. Bobot pada *hidden layer* dan *output layer* level campuran terdapat pada Tabel 3.13 dan Tabel 3.14, sedangkan bobot pada *hidden layer* dan *output layer* level minyak terdapat pada Tabel 3.15 dan Tabel 3.16.

Tabel 3. 13 Bobot antara *input layer* dan *hidden layer* Level Campuran

<i>Hidden Weight Level Campuran</i>					
	z_1	z_2	z_3	z_4	z_5
x_1	-0.25	-1.40	1.52	-1.59	-1.41
x_2	0.07	-0.83	0.39	-0.21	1.17
x_3	0.21	1.02	-0.73	1.13	-0.78
x_4	0.18	-0.88	0.32	-0.24	1.14
x_5	0.13	1.29	-0.70	0.91	-0.67
x_6	0.21	-0.08	0.34	0.54	0.68
x_7	0.21	-0.29	-0.50	-0.03	-0.43
x_8	-0.57	-0.06	-0.31	0.01	-0.29

Tabel 3. 14 Bobot antara *hidden layer* dan *output layer* Level Campuran

Output Weight Level Campuran			
	y_1	y_2	y_3
z_1	0.24	-0.17	-0.40
z_3	2.46	-0.24	0.34
z_2	-1.29	-1.87	-0.78
z_4	2.08	0.75	-0.53
z_5	-0.6	-1.17	-2.30

Tabel 3. 15 Bobot antara *input layer* dan *hidden layer* Level Minyak

Hidden Weight Level Minyak					
	z_1	z_2	z_3	z_4	z_5
x_1	0.33	1.16	0.56	1.96	-0.54
x_2	-0.07	0.83	0.55	-0.74	-0.68
x_3	-0.05	-0.86	-0.98	-0.49	0.12
x_4	-0.16	0.71	0.59	-0.68	-0.59
x_5	-0.52	-0.94	-0.84	-0.38	0.28
x_6	-0.08	1.03	-0.52	0.19	0.53
x_7	-0.17	-0.43	0.68	-0.5	-0.7
x_8	-0.31	-0.24	0.41	-0.58	-1.09

Tabel 3. 16 Bobot antara *input layer* dan *hidden layer* Level Minyak

Output Weight Level Minyak			
	y_1	y_2	y_3
z_1	-0.2	-0.14	-0.11
z_3	-0.2	-1.46	-1.19
z_2	-0.4	-1.45	1.02
z_4	-1.56	-0.41	-1.21
z_5	0.73	0.46	-1.3

Halaman ini sengaja dikosongkan

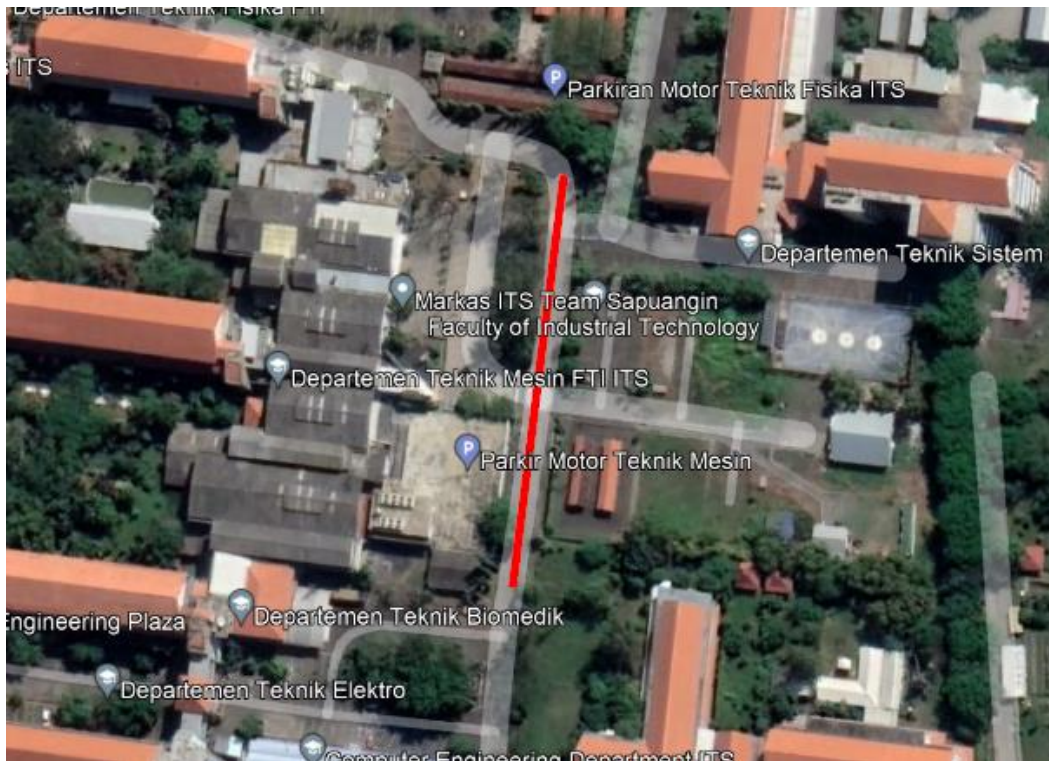
BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan terkait hasil pengujian serta analisis dari metode yang telah dilakukan pada bab sebelumnya. Pengujian yang dilakukan meliputi analisis performansi sistem komunikasi IoT, uji performansi sistem pengendalian PID, dan uji performansi sistem pengendalian *Neural Network*-PID.

4.1 Analisis Hasil Pengujian Sistem Komunikasi *Wireless Sensor and Actuator Networks*

Pengujian sistem komunikasi *Wireless Sensor and Actuator Networks* (WSAN) dilakukan dalam propagasi *Line of Sight* (LoS). Dimana LoS didefinisikan antara *receiver* dan *transceiver* tidak terdapat penghalang. Pengujian dilakukan dengan jarak 100 meter dengan mengirimkan *20 bytes data dummy*. Kontrol *slave* bekerja mengirim data menuju *master*, apabila register dari kedua perangkat antara *slave* dan *master* sesuai maka dapat berkomunikasi. Lokasi pengambilan data berdasarkan Gambar 4.1 dan hasil uji sistem komunikasi *wireless sensor and actuator network* ditampilkan pada Tabel 4.1.



Gambar 4. 1 Lokasi Pengambilan Data Sistem Komunikasi

Tabel 4. 1 Performansi Komunikasi Wireless

RSSI	SNR	Time Transmitted (s)	Time Received (s)	Delay Time (s)
-61	9.75	02:18:00.703	02:18:00.564	00:00:00.139
-69	9.75	02:18:07.515	02:18:07.373	00:00:00.142
-58	9.5	02:18:14.191	02:18:14.032	00:00:00.159
-65	9.5	02:18:36.345	02:18:36.208	00:00:00.137
-67	10	02:18:40.455	02:18:40.315	00:00:00.140
-57	9.5	02:18:49.831	02:18:49.686	00:00:00.145
-49	9.5	02:18:56.078	02:18:55.935	00:00:00.143
-50	9.5	02:19:02.750	02:19:02.604	00:00:00.146
-58	9.75	02:19:16.378	02:19:16.223	00:00:00.155

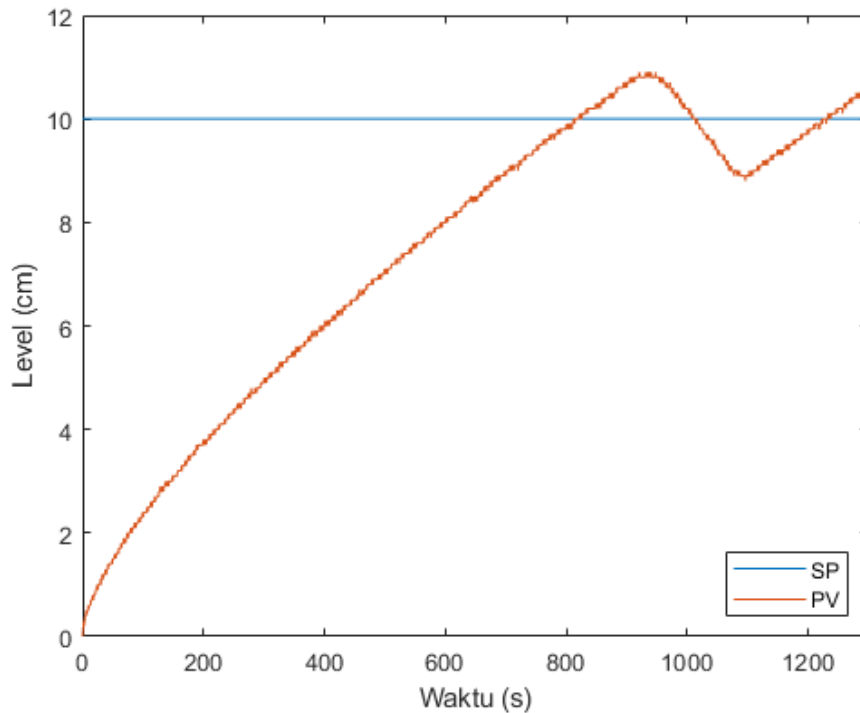
Berdasarkan hasil pengujian pada Tabel 4.1, sistem komunikasi *wireless sensor and actuator networks* memiliki *delay time* rata-rata 145 ms. Nilai tersebut masih memenuhi standar yang telah disebutkan sebelumnya terkait rekomendasi *delay time* yaitu 150 ms (Union, 2003).

4.2 Analisis Performansi Sistem Pengendalian PID

Pengujian dilakukan untuk melihat karakteristik dinamik dari *plant*. Penentuan nilai awal parameter PID sudah dilakukan berdasarkan pada sub bab 3.5. Pengujian performa sistem pengendali PID dilakukan selama 1300 detik pada campuran dan minyak dengan *set point* yang telah ditetapkan 10 cm.

4.2.1 Sistem Pengendalian PID Level Campuran

Berdasarkan hasil pada sub bab 3.5, setelah dilakukan pengujian *open loop* yang kemudian parameter PID ditentukan melalui pendekatan gradien, didapatkan nilai 3.8644 untuk Kp, 0.8093 untuk Ki, 1.4762 dan untuk Kd. Hasil respon dinamik dari gain PID ini ditampilkan pada Gambar 4.2, dimana variabel proses masih belum mencapai keadaan *settling* dengan *error steady-state* yang belum memenuhi kriteria.



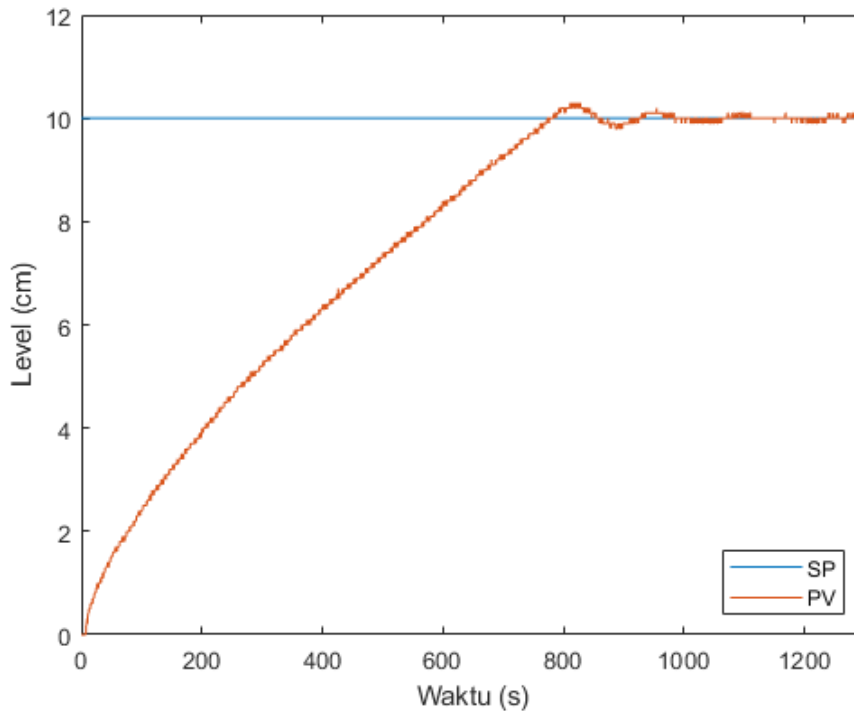
Gambar 4. 2 Respon Dinamik Sistem Pengendalian PID Level Campuran

Hasil karakteristik respon dinamik yang dihasilkan pada sistem pengendali PID level campuran ditampilkan pada Tabel 4.2.

Tabel 4. 2 Respon Dinamik Level Campuran dengan Sistem Pengendalian PID

Kriteria Performansi	Nilai
Rise time (s)	677.024 s
Settling Time (s)	NULL
Max. overshoot (%)	9
Error Steady State (%)	6

Berdasarkan hasil pada Tabel 4. 2, respon dinamik belum mencapai keadaan *steady* dimana hasil dari *tuning* PID setelah uji *open loop* belum memenuhi kriteria yang diharapkan, maka dilakukan *fine tuning*. *Fine tuning* dilakukan dengan parameter PID 45 untuk Kp, 5 untuk Ki, dan 15 untuk Kd. Hasil respon dinamik yang dihasilkan tiap kombinasi parameter PID tersebut ditunjukkan pada Gambar 4.3.



Gambar 4. 3 Respon Dinamik Sistem Pengendalian PID Level Campuran Setelah *Fine Tuning*

Hasil karakteristik respon dinamik yang dihasilkan pada sistem pengendali PID level campuran ditampilkan pada Tabel 4.3.

Tabel 4. 3 Respon Dinamik Level Minyak dengan Sistem Pengendalian PID Setelah *Fine Tuning*

Kriteria Performansi	Nilai
Rise time (s)	642.709
Settling Time (s)	828.921
Max. overshoot (%)	3
Error Steady State (%)	0

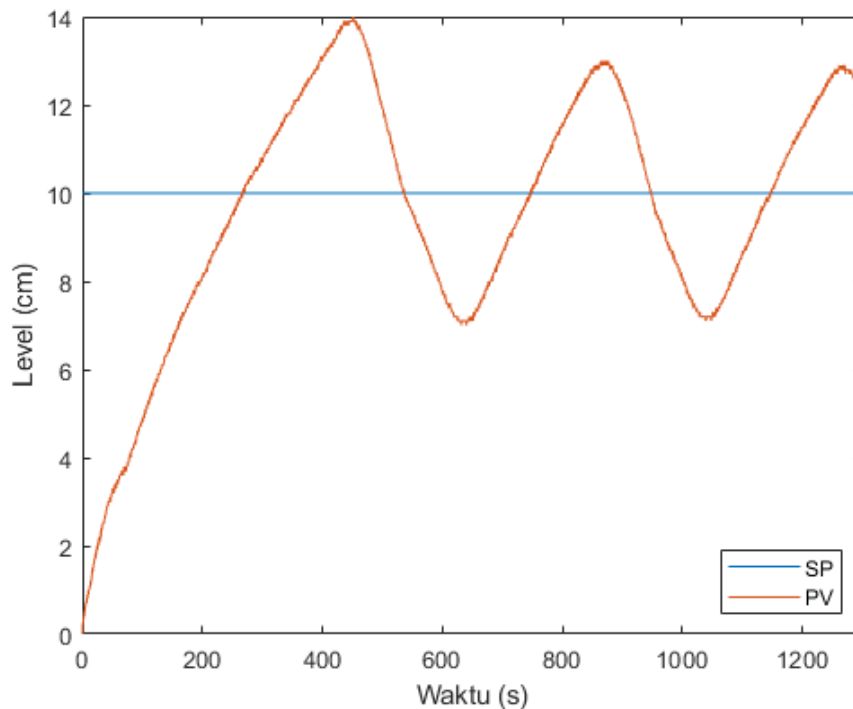
Berdasarkan hasil percobaan setelah *fine tuning*, respon dinamik pada *plant* lebih baik dengan ditandai respon dinamik mencapai keadaan *settling* dengan *error steady state* senilai 0%. Percobaan juga dilakukan dengan *tuning* manual PID dengan kombinasi parameter PID lainnya yang terlampir pada lampiran C. Hasil percobaan lainnya terlampir pada Tabel 4.4

Tabel 4. 4 Respon Dinamik Sistem Pengendalian PID Level Campuran dengan Variasi Kombinasi Kp, Ki, dan Kd

Variasi (Kp,Ki,Kd)	Rise Time (s)	Settling Time (s)	Max. Overshoot (%)	Error Steady State (%)
C1 (65,20,3)	629.572	739.378	2	1
C2 (95,10,15)	620.083	826.988	4	0

4.2.2 Sistem Pengendalian PID Level Minyak

Berdasarkan pengujian *open loop* yang kemudian parameter PID ditentukan melalui pendekatan gradien, didapatkan nilai 1.170 untuk Kp, 0.1721 untuk Ki, 0.285 dan untuk Kd. Hasil respon dinamik dari pengendali PID dengan gain yang sudah dikalkulasikan ini ditampilkan pada Gambar 4.2. Variabel proses yang dihasilkan masih belum mencapai keadaan *settling* dengan *overshoot* mencapai 40% dan *error steady-state* yang belum memenuhi kriteria.



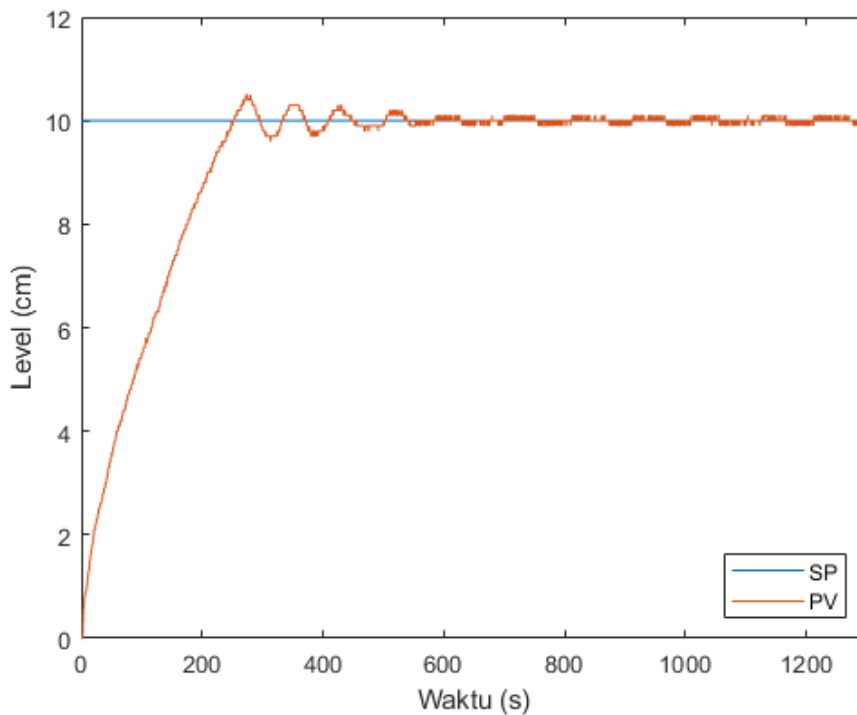
Gambar 4. 4 Respon Dinamik Pengendali PID Level Minyak

Hasil karakteristik respon dinamik yang dihasilkan pada sistem pengendali PID level minyak ditampilkan pada Tabel 4.5

Tabel 4. 5 Respon Dinamik Level Minyak dengan Pengendali PID

Kriteria Performansi	Nilai
Rise time (s)	219.567
Settling Time (s)	NULL
Max. overshoot (%)	40
Error Steady State (%)	6

Berdasarkan hasil pada Tabel 4. 4, hasil dari *tuning* PID sama seperti pada level campuran dimana hasil uji *open loop* belum memenuhi kriteria yang diharapkan dengan error steady state kurang dari sama dengan 5% dan belum mencapai kondisi *settling*, sehingga diperlukan *fine tuning*. *Fine tuning* dilakukan dengan parameter PID 18 untuk Kp, 6 untuk Ki, dan 20 untuk Kd. Respon dinamik yang dihasilkan setelah dilakukan *tuning* ulang terdapat pada Gambar 4.5

**Gambar 4. 5** Respon Dinamik Sistem Pengendalian PID Level Minyak Setelah *Fine Tuning*

Hasil karakteristik respon dinamik yang dihasilkan setelah dilakukan *fine tuning* pada sistem pengendali PID level minyak ditampilkan pada Tabel 4.6

Tabel 4. 6 Respon Dinamik Level Minyak dengan Sistem Pengendalian PID Setelah *Fine Tuning*

Kriteria Performansi	Nilai
Rise time (s)	222.391
Settling Time (s)	442.075
Max. overshoot (%)	5
Error Steady State (%)	1

Respon dinamik pada *plant* setelah *fine tuning* lebih baik dibandingkan hasil tuning parameter dari uji *open loop* dengan ditandai respon dinamik mencapai keadaan *settling* dengan *error steady state* senilai 1%. Percobaan juga dilakukan dengan *tuning* manual PID dengan kombinasi parameter PID lainnya yang terlampir pada lampiran D. Hasil percobaan lainnya terlampir pada Tabel 4.7.

Tabel 4. 7 Respon Dinamik Sistem Pengendalian PID Level Minyak dengan Variasi Kombinasi Kp, Ki, dan Kd

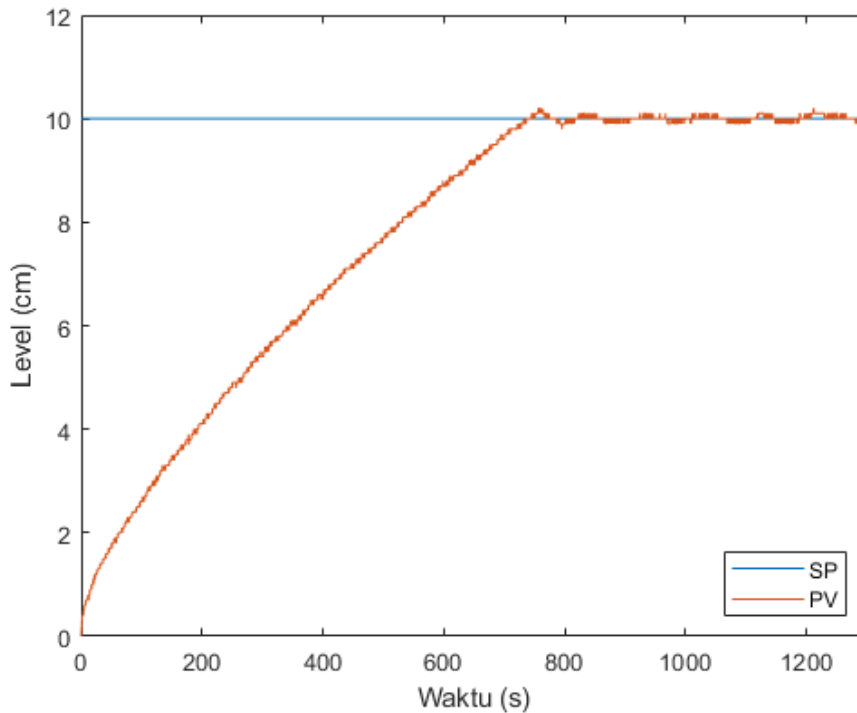
Variasi (Kp,Ki,Kd)	Rise Time (s)	Settling Time (s)	Max. Overshoot (%)	Error Steady State (%)
D1 (24,15,30)	226.986	373.701	3	1
D2 (35,15,10)	208.991	NaN	3	3

4.3 Analisis Performansi Sistem Pengendalian *Neural Network* PID

Pengujian dilakukan untuk melihat karakteristik dari *plant* setelah dilakukan pelatihan *neural network* menggunakan *backpropagation*. Hasil dari pelatihan ini memiliki bobot pada *hidden layer* dan *output layer* yang dimasukkan kedalam *neural network feed forward* sehingga mengikuti kebutuhan *output* yang diharapkan berdasarkan data *input output* yang ada. Sistem pengujian *Neural Network* PID ini diharapkan dapat memperbaiki kekurangan yang ada pada pengendali PID konvensional. Pengujian dilakukan selama 1300 detik dengan setpoint antara level campuran dan minyak sama yaitu 10 cm.

4.3.1 Sistem Pengendalian *Neural Network* PID Level Campuran

Berdasarkan *state* yang telah dirancang dengan target *output* pada *neural network* pada Lampiran A untuk level campuran, didapatkan Hasil respon dinamik sistem pengendalian *Neural Network* PID ditampilkan pada Gambar 4.6.



Gambar 4. 6 Respon Dinamik Sistem Pengendalian NN-PID Level Campuran

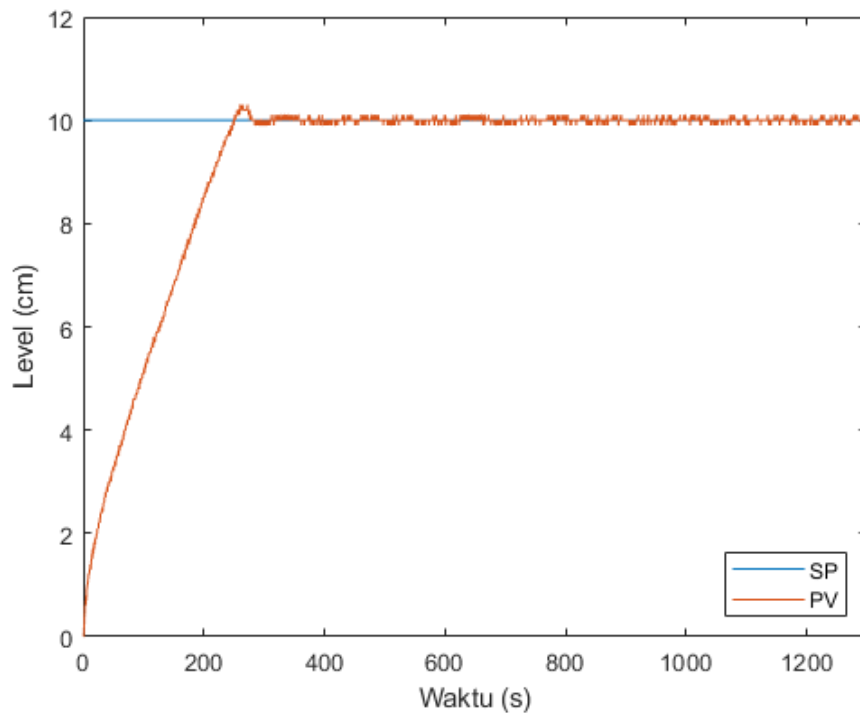
Hasil karakteristik respon dinamik yang dihasilkan saat menggunakan pengendali *Neural Network* PID level campuran ditampilkan pada Tabel 4.8.

Tabel 4. 8 Respon Dinamik Level Campuran dengan Sistem Pengendalian *Neural Network* PID

Kriteria Performansi	Nilai
Rise time (s)	617.198
Settling Time (s)	709.933
Max. overshoot (%)	2
Error Steady State (%)	1

4.3.2 Sistem Pengendalian *Neural Network* PID Level Minyak

Berdasarkan *state* yang telah dirancang dengan target *output* pada *neural network* pada Lampiran A untuk level minyak, didapatkan hasil respon dinamik pengendali *Neural Network* PID ditampilkan pada Gambar 4.7.



Gambar 4. 7 Respon Dinamik Sistem Pengendalian NN-PID Level Minyak

Hasil karakteristik respon dinamik yang dihasilkan saat menggunakan pengendali *Neural Network* PID level minyak ditampilkan pada Tabel 4.9.

Tabel 4. 9 Respon Dinamik Level Minyak dengan Sistem Pengendali *Neural Network* PID

Kriteria Performansi	Nilai
Rise time (s)	210.218
Settling Time (s)	272.649
Max. overshoot (%)	3
Error Steady State (%)	1

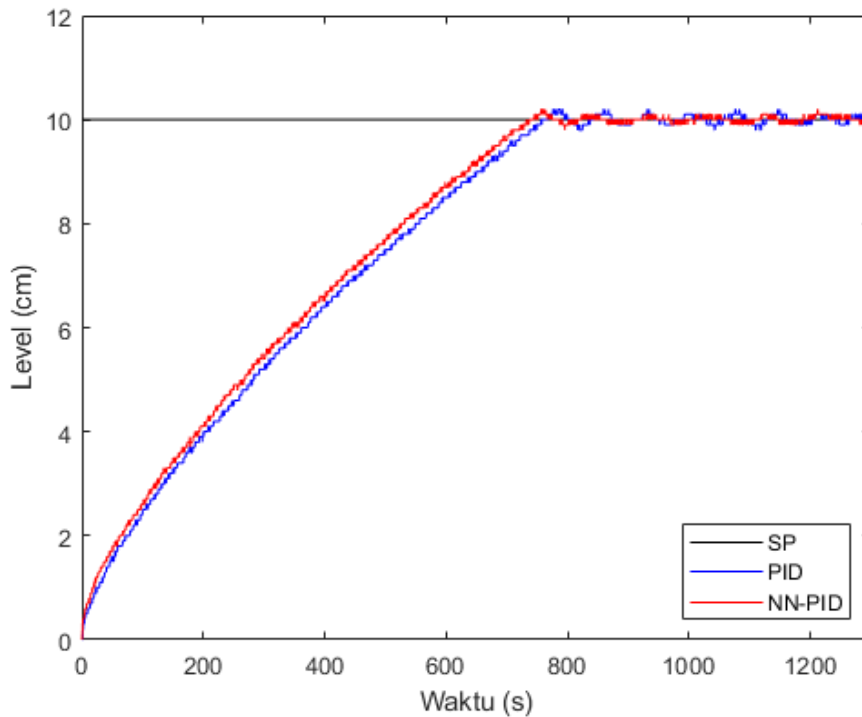
4.4 Analisis Perbandingan Sistem Pengendalian PID dan *Neural Network* PID

Perbandingan hasil uji dilakukan dengan membandingkan performansi sistem pengendalian *Neural Network* PID dengan pengendali PID yang telah dilakukan *tuning* dengan hasil paling baik.

4.4.1 Perbandingan Hasil Uji Level Campuran

Hasil uji pengendali PID yang terbaik diambil kemudian dibandingkan dengan hasil uji *Neural Network* PID. Hasil uji yang diambil dalam pengendali PID campuran ialah dengan parameter K_p dengan nilai 65, K_i dengan nilai 20, dan K_d dengan nilai 3. Pemilihan ini dilakukan karena memiliki *rise time* yang cukup baik dibandingkan dengan lainnya, serta

memiliki *error steady state* dan *overshoot* maksimum yang rendah. Hasil uji dari performansi dibandingkan pada Gambar 4.8.



Gambar 4. 8 Perbandingan Respon Sistem Pengendalian PID dan *Neural Network* PID Pada Level Campuran

Hasil analisis kedua sistem pengendalian dari respon dinamik dibandingkan yang ditunjukkan pada Tabel 4.10.

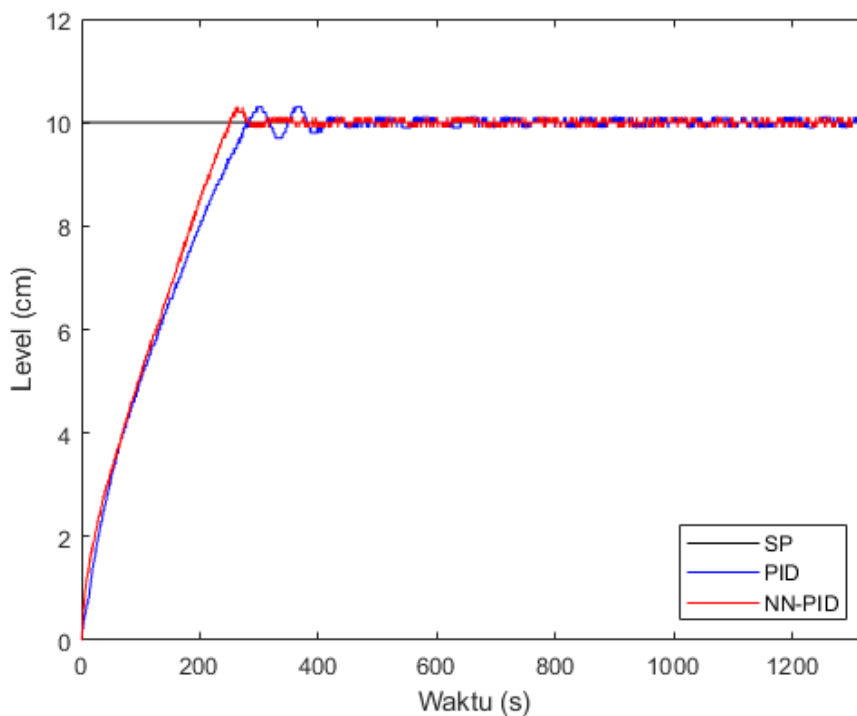
Tabel 4. 10 Perbandingan Respon Dinamik Sistem Pengendalian PID dan *Neural Network* PID Pada Level Campuran

Kriteria Performansi	<i>Neural Network</i> PID	PID
Rise time (s)	617.198	629.572
Settling Time (s)	709.933	739.378
Max. overshoot (%)	2	2
Error Steady State (%)	1	1

Dari hasil yang didapatkan pada Tabel 4.10 menunjukkan bahwa pengendalian *Neural Network* PID memiliki *rise time* 12.374 detik lebih baik dibandingkan pengendalian PID dan *settling time* lebih rendah 29.445 detik dibandingkan pengendalian PID. Maksimum overshoot yang dihasilkan serta *error steady state* tidak terdapat perbandingan. Performansi pengendalian *Neural Network* PID menghasilkan perubahan yang lebih baik terhadap performansi pengendalian PID.

4.4.2 Perbandingan Hasil Uji Level Minyak

Hasil uji pengendali PID level minyak yang terbaik diambil kemudian dibandingkan dengan hasil uji *Neural Network* PID. Hasil uji yang diambil dalam pengendali PID minyak ialah dengan parameter K_p dengan nilai 24, K_i dengan nilai 15, dan K_d dengan nilai 30. Pemilihan ini dilakukan karena memiliki waktu *settling* yang cepat dibandingkan dengan lainnya, serta memiliki *error steady state* dan *overshoot* maksimum yang rendah. Hasil uji dari performansi dibandingkan pada Gambar 4.9.



Gambar 4. 9 Perbandingan Respon Sistem Pengendalian PID dan *Neural Network* PID Pada Level Minyak

Hasil analisis kedua sistem pengendalian dari respon dinamik dibandingkan yang ditunjukkan pada Tabel 4.11.

Tabel 4. 11 Perbandingan Respon Dinamik Sistem Pengendalian PID dan *Neural Network* PID Pada Level Minyak

Kriteria Performansi	<i>Neural Network</i> PID	PID
Rise time (s)	210.218	226.986
Settling Time (s)	272.649	373.701
Max. overshoot (%)	3	3
Error Steady State (%)	1	1

Dari hasil yang didapatkan pada Tabel 4.11 menunjukkan bahwa pengendalian *Neural Network* PID level minyak juga memiliki memiliki performansi respon yang lebih baik dibandingkan PID. Dimana *rise time Neural Network* PID 16.768 detik lebih baik dibandingkan pengendalian PID dan *settling time* lebih rendah 101.052 detik yang lebih baik dari pengendalian PID. Kedua sistem pengendalian ini memiliki juga tidak memiliki perbedaan pada *error steady state* serta pada maksimum *overshoot*. Perbedaan dari kedua respon dinamik pengendalian level minyak memiliki kelebihan yang sama seperti penerapan sistem pengendalian *Neural Network* PID pada Level Campuran.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari data penelitian yang telah dilakukan mengenai perancangan pengendali *Neural Network* PID pada separator tiga fasa berbasis *Internet of Things*, maka dapat disimpulkan bahwa:

- Pengendali adaptif *Neural Network* PID memiliki performansi yang lebih baik dalam pengendalian level campuran dan minyak di *mini plant* separator tiga fasa dibandingkan dengan pengendali PID konvensional. Pada pengendalian level campuran respon dinamik *plant* menghasilkan selisih *rise time* 12.374 detik lebih rendah dan *settling time* lebih rendah 29.445 detik dari sistem pengendalian PID. Pada pengendalian level minyak respon dinamik *plant* menghasilkan selisih *rise time* 16.768 detik lebih rendah dan *settling time* lebih rendah 101.052 detik dari sistem pengendalian PID. Kedua sistem pengendalian tidak memiliki perbedaan pada *overshoot* maksimum dan *error steady state*.
- Sistem komunikasi *wireless sensor and actuator networks* dengan menggunakan perangkat LoRa pada separator tiga fasa mampu berkomunikasi secara *realtime* dengan *delay time* rata-rata 145 ms pada jarak 100 meter.

5.2 Saran

Adapun saran yang diberikan oleh penulis untuk pengembangan selanjutnya mengenai penelitian ini sebagai berikut.

- Pengembangan sistem pengendalian pada *mini plant* juga melakukan pengendalian pada aliran masuk separator dengan mengatur kendali tegangan pada pompa masukan sehingga dapat meningkatkan performansi respon dinamik *plant*.
- Perancangan sistem pengendalian level dapat dikembangkan menggunakan sistem pengendalian adaptif PID berbasis *neural network* lainnya atau menggunakan algoritma lain seperti sistem kendali *Neural Network Controller* atau ANFIS.
- Sistem komunikasi berbasis *Internet of Things* pada *mini plant* dapat dikembangkan menggunakan *Narrow Band IoT* (NB-IoT) untuk menghasilkan sistem komunikasi dengan *data rate* yang lebih tinggi serta penggunaan biaya operasional yang lebih rendah.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials*, 17(4), 2347–2376. <https://doi.org/10.1109/COMST.2015.2444095>
- Bosch. (2015). *BMP 180 Digital Pressure Sensor*. <https://www.bosch-sensortec.com/>
- Chairuzzaini, Rusli, M., & Ariyanto, R. (1998). *Pengenalan Metode Ziegler-Nichols pada Perancangan Kontroler pada PID*. Elektroindonesia.Com. <https://www.elektroindonesia.com/elektro/tutor12.html>
- Chen, J., Díaz, M., Llopis, L., Rubio, B., & Troya, J. M. (2011). A survey on quality of service support in wireless sensor and actor networks: Requirements and challenges in the context of critical infrastructure protection. *Journal of Network and Computer Applications*, 34(4), 1225–1239. <https://doi.org/10.1016/j.jnca.2011.01.008>
- Circuits4you. (2018). *ESP32 DevKit ESP32-WROOM GPIO Pinout*. Circuits4you.Com. <https://circuits4you.com/2018/12/31/esp32-devkit-esp32-wroom-gpio-pinout/>
- Components101. (2019). *MG996R Servo Motor*. Components101.Com. <https://components101.com/motors/mg996r-servo-motor-datasheet>
- Components101. (2021). *MS5611 Atmospheric Pressure & Temperature Sensor Module*. <https://Components101.Com/>. <https://components101.com/sensors/ms5611-atmospheric-pressure-temperature-sensor-module>
- da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., & dos Reis Alves, S. F. (2017). Artificial Neural Network Architectures and Training Processes. In *Artificial Neural Networks: A Practical Course* (pp. 21–28). Springer International Publishing. <https://doi.org/10.1007/978-3-319-43162-8>
- Dataforth Corporation. (2020). *Tuning Level Control Loops*.
- Dejan. (2020). *How to Control Servo Motors with Arduino*. How To Mechatronics. <https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>
- Diligent Dev. (2020). *Firebase Cloud Functions Tutorial — Creating a REST API*. Avascript.Plainenglish.Io. <https://javascript.plainenglish.io/firebase-cloud-functions-tutorial-creating-a-rest-api-8cbc51479f80>
- Emerson Fisher. (2005). Control Valve Handbook. In *Emerson Automation Solutions* (5th ed.).

- Espressif Systems. (2021). *ESP32 Series Datasheet*. Espressif.Com. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- Fausett, L. V. (1993). *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*. Pearson Education Inc.
- Geng, H. (2017). Internet of Things and data analytics handbook. In *Internet of Things and Data Analytics Handbook*. <https://doi.org/10.1002/9781119173601>
- Ghaffarkhah, A., Ameri Shahrabi, M., Keshavarz Moraveji, M., & Eslami, H. (2017). Application of CFD for designing conventional three phase oilfield separator. *Egyptian Journal of Petroleum*, 26(2), 413–420. <https://doi.org/10.1016/j.ejpe.2016.06.003>
- Goodwin, G. C., Graebe, S. F., & Salgado, M. E. (2000). Control Systems Design. In *Eshbach's Handbook of Engineering Fundamentals, Fifth Edition* (1st ed.). Pearson Education Inc. <https://doi.org/10.1002/9780470432754.ch13>
- Grilo, A. M., Chen, J., Diaz, M., Garrido, D., & Casaca, A. (2014). An integrated WSN and SCADA system for monitoring a critical infrastructure. *IEEE Transactions on Industrial Informatics*, 10(3), 1755–1764. <https://doi.org/10.1109/TII.2014.2322818>
- Gude, J. J., Kahoraho, E., & Etxaniz, J. (2006). Practical aspects of PID controllers: An industrial experience. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, February 2021*, 870–878. <https://doi.org/10.1109/ETFA.2006.355215>
- Havard, D. (2013). *Oil and Gas Production Handbook - An introduction to oil and gas production, transport, refining and petrochemical industry*. ABB Oil and Gas.
- Haykin, S. (1999). *Neural Networks and Learning Machines: Third Edition*. Pearson Prentice Hall.
- HOPERF. (2020). *RFM95/96/97/98(W) - Low Power Long Range Transceiver Module*. <https://www.hoperf.com/data/upload/portal/20190801/RFM95W-V2.0.pdf>
- Hübschmann, I. (2020). *ESP32 for IoT: A Complete Guide*. Nabto.Com. <https://www.nabto.com/guide-to-iot-esp-32/>
- Ismail, A. N., Prajitno, P., & Adhitya, K. T. (2021). Application of the adaptive neuro-fuzzy inference system (ANFIS) for simulating water fluid level control systems on horizontal separator. *AIP Conference Proceedings*, 2374(ISC PMS 2020), 1–9. <https://doi.org/10.1063/5.0059111>
- King, M. (2011). Process Control: A Practical Approach. In *Process Control: A Practical Approach* (1st ed.). John Wiley and Sons. <https://doi.org/10.1002/9780470976562>
- Kominfo. (2018). *Rancangan Peraturan Menteri Komunikasi dan Informatika Republik*

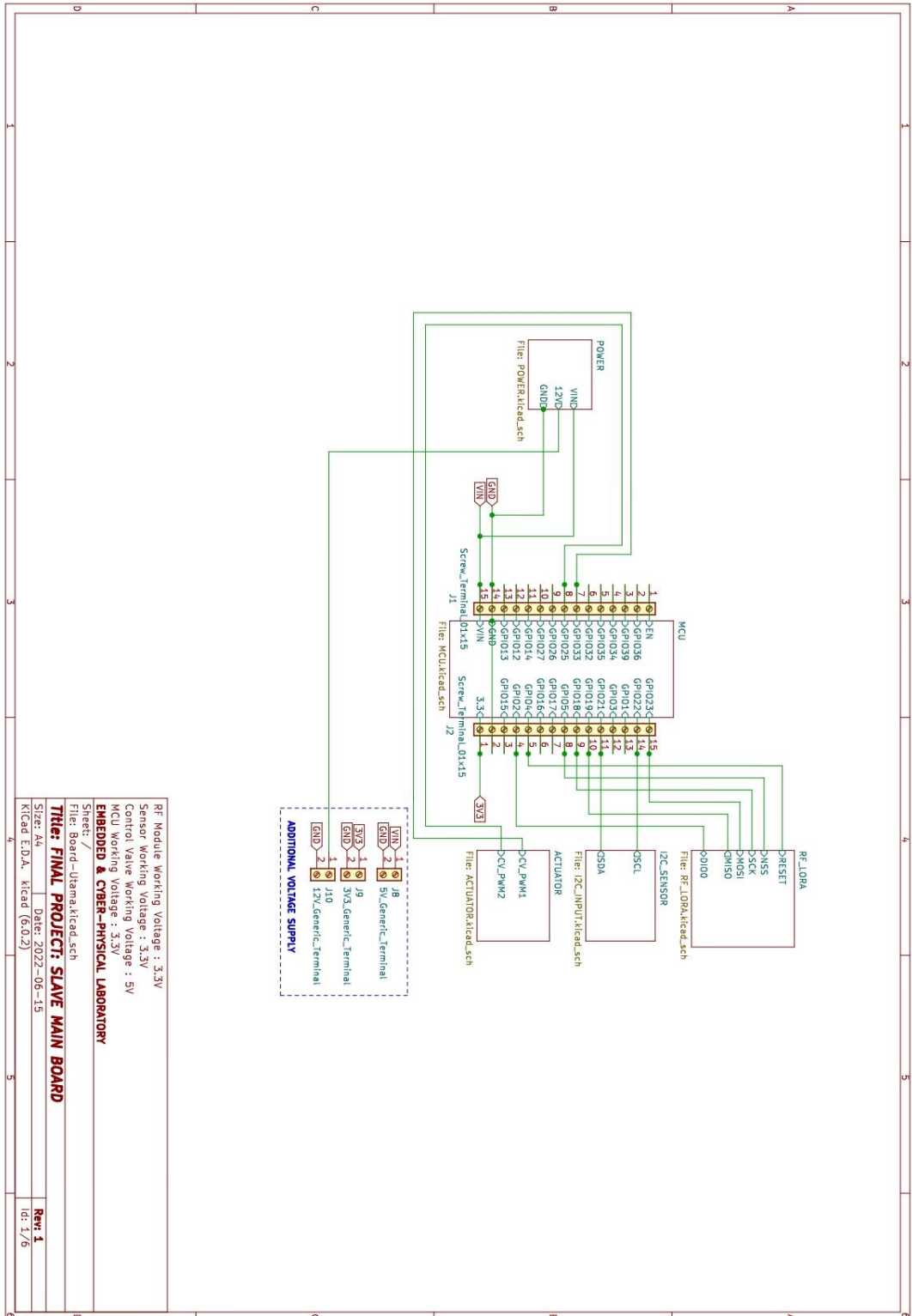
Indonesia : Persyaratan Teknis Alat dan / atau Perangkat Telekomunikasi Low Power Wide Area.

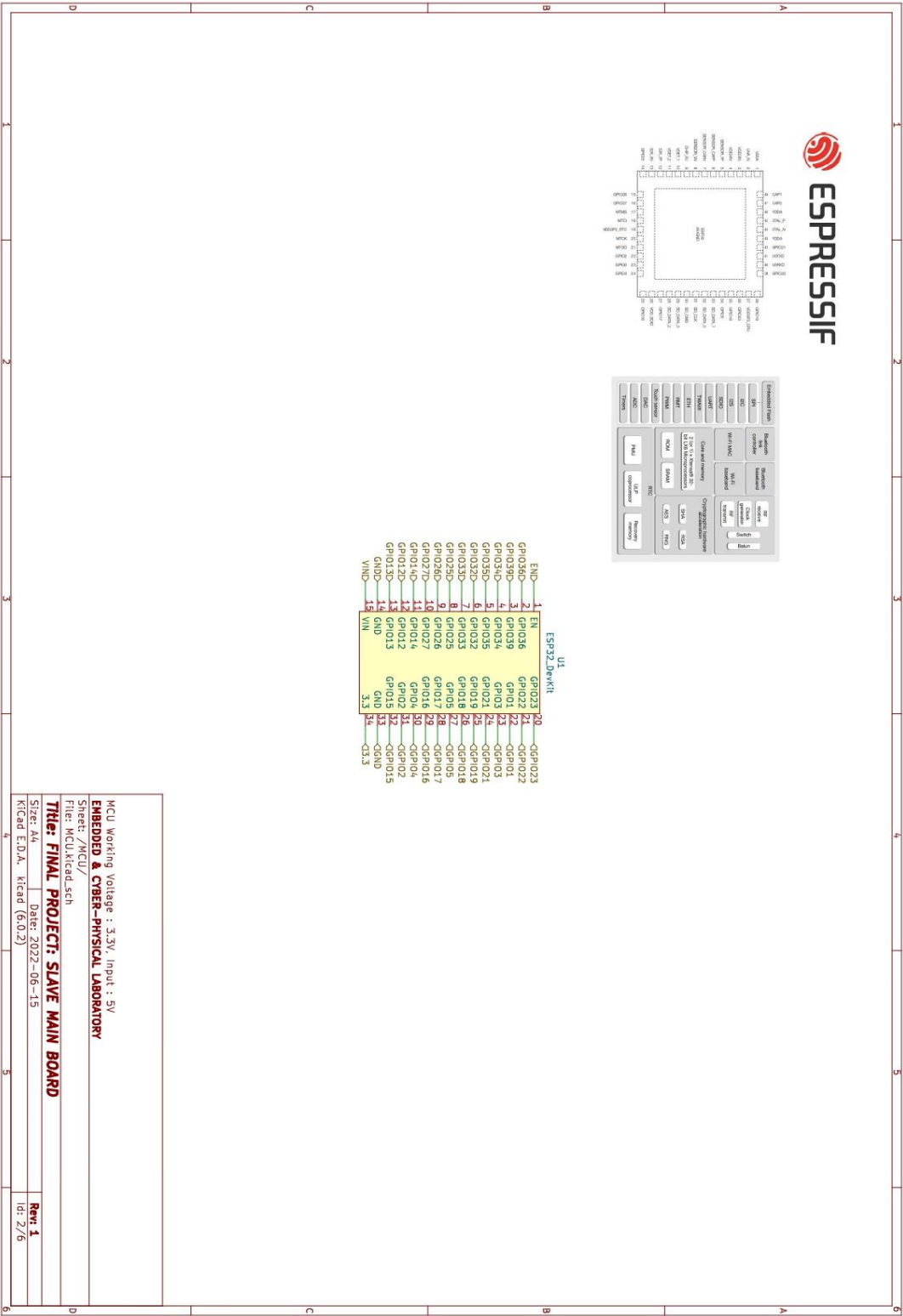
- Kukreja, H., N, B., C S, S., & S, K. (2016). An introduction to Artificial Neural Network. *International Journal Of Advance Research And Innovative Ideas In Education*, 1(5), 1–5. <https://doi.org/10.2514/6.1994-294>
- Li, Y., Ang, K. H., & Chong, G. C. Y. (2006). PID Control System Analysis and Design: Problems, Remedies, and Future Directions. *IEEE Control Systems*, 26(1), 32–41. <https://doi.org/10.1109/MCS.2006.1580152>
- Liang, H., Sang, Z. K., Wu, Y. Z., Zhang, Y. H., & Zhao, R. (2021). High precision temperature control performance of a PID neural network-controlled heater under complex outdoor conditions. *Applied Thermal Engineering*, 195(December 2020), 117234. <https://doi.org/10.1016/j.applthermaleng.2021.117234>
- Lombardi, M., Pascale, F., & Santaniello, D. (2021). Internet of Things: A general overview between architectures, protocols and applications. *Information (Switzerland)*, 12(2), 1–21. <https://doi.org/10.3390/info12020087>
- Moroney, L. (2017). *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform* (1st ed.). Apress.
- Negnevitsky, M. (2005). *Artificial Intelligence : A Guide to Intelligent Systems* (2nd ed.). Pearson Education Inc.
- Ogata, K. (2010). *Modern Control Engineering* (5th ed., Vol. 93, Issue 1). Pearson Education Inc. <https://doi.org/10.1115/1.3426465>
- Oilprocessing. (2019). *Three-phase Oil and Gas Separation - Chapter 3*. <https://oilprocessing.net>. <https://oilprocessing.net/oil/viewtopic.php?t=9>
- Parker, J. (2016). *Servo Valve from Ball Valve + Hobby Servo*. Thingiverse. <https://www.thingiverse.com/thing:1659407>
- Priyanka, E. B., Maheswari, C., Thangavel, S., & Bala, M. P. (2020). Integrating IoT with LQR-PID controller for online surveillance and control of flow and pressure in fluid transportation system. *Journal of Industrial Information Integration*, 17(January), 100127. <https://doi.org/10.1016/j.jii.2020.100127>
- Righetti, F., Vallati, C., Comola, D., & Anastasi, G. (2019). Performance Measurements of IEEE 802.15.4g Wireless Networks. *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 1–6. <https://doi.org/10.1109/WoWMoM.2019.8793051>
- Semtech. (2020). LoRa and LoRaWAN: A Technical Overview. In *Semtech*. Semtech.

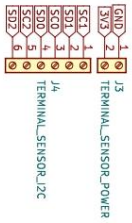
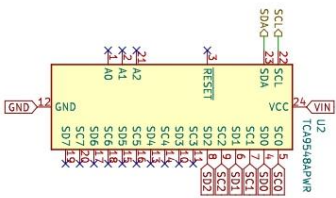
- <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>
- Septalita, E. D. (2018). KECELAKAAN KERJA DI AREA PENGEBORAN MINYAK DAN GAS TAHUN 2012. *The Indonesian Journal of Occupational Safety and Health*, 7(1), 52. <https://doi.org/http://dx.doi.org/10.20473/ijosh.v7i1.2018.52-62>
- Serpanos, D., & Wolf, M. (2017). Internet-of-things (IoT) systems: Architectures, algorithms, methodologies. In *Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies*. <https://doi.org/10.1007/978-3-319-69715-4>
- Subakti, A. (2021). *IMPLEMENTASI SISTEM KONTROL FUZZY PID BERBASIS INDUSTRIAL INTERNET OF THINGS PADA MINI PLANT SEPARATOR TIGA FASA*. Institut Teknologi Sepuluh Nopember.
- Sudrajat, S. (2020). *RANCANG BANGUN SISTEM PENGENDALIAN FUZZY PI PADA MINI PLANT SEPARATOR TIGA FASA MENGGUNAKAN INTERFACE DCS BERBASIS WIRELESS SENSOR NETWORK DAN IOT SECARA REAL-TIME*. Institut Teknologi Sepuluh Nopember.
- Syauqi, M. (2009). *Perancangan Sistem Pengendalian Level Minyak Pada Separator (PV 9900) PT.JOB Pertamina-Petrochina Tuban Dengan Tuning Parameter Kontroler Proportional Integral Menggunakan Metode Direct Synthesis*. Institut Teknologi Sepuluh Nopember.
- TE Connectivity Ltd. (2017). *MS5611-01BA03 Barometric Pressure Sensor*. Te.Com. https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5611-01BA03%7FB3%7Fpdf%7FEnglish%7FENG_DS_MS5611-01BA03_B3.pdf%7FCAT-BLPS0036
- Union, I. T. (2003). *One-way Transmission Time*.
- Wei, Y. (2018). Study of Double-holding Water Tank Liquid Level Control Base on Neural Networks PID Control. *International Conference on Computer Modeling, Simulation and Algorithm (CMSA 2018)*, 151(Cmsa), 320–323. <https://doi.org/10.2991/cmsa-18.2018.73>
- Yang, Y. (2020). Comparison of Various PID Control Algorithms on Coupled-Tank Liquid Level Control System. *Journal of Physics: Conference Series*, 1622(1). <https://doi.org/10.1088/1742-6596/1622/1/012129>

LAMPIRAN

Lampiran A. Wiring Diagram Sistem Kontrol *Slave*

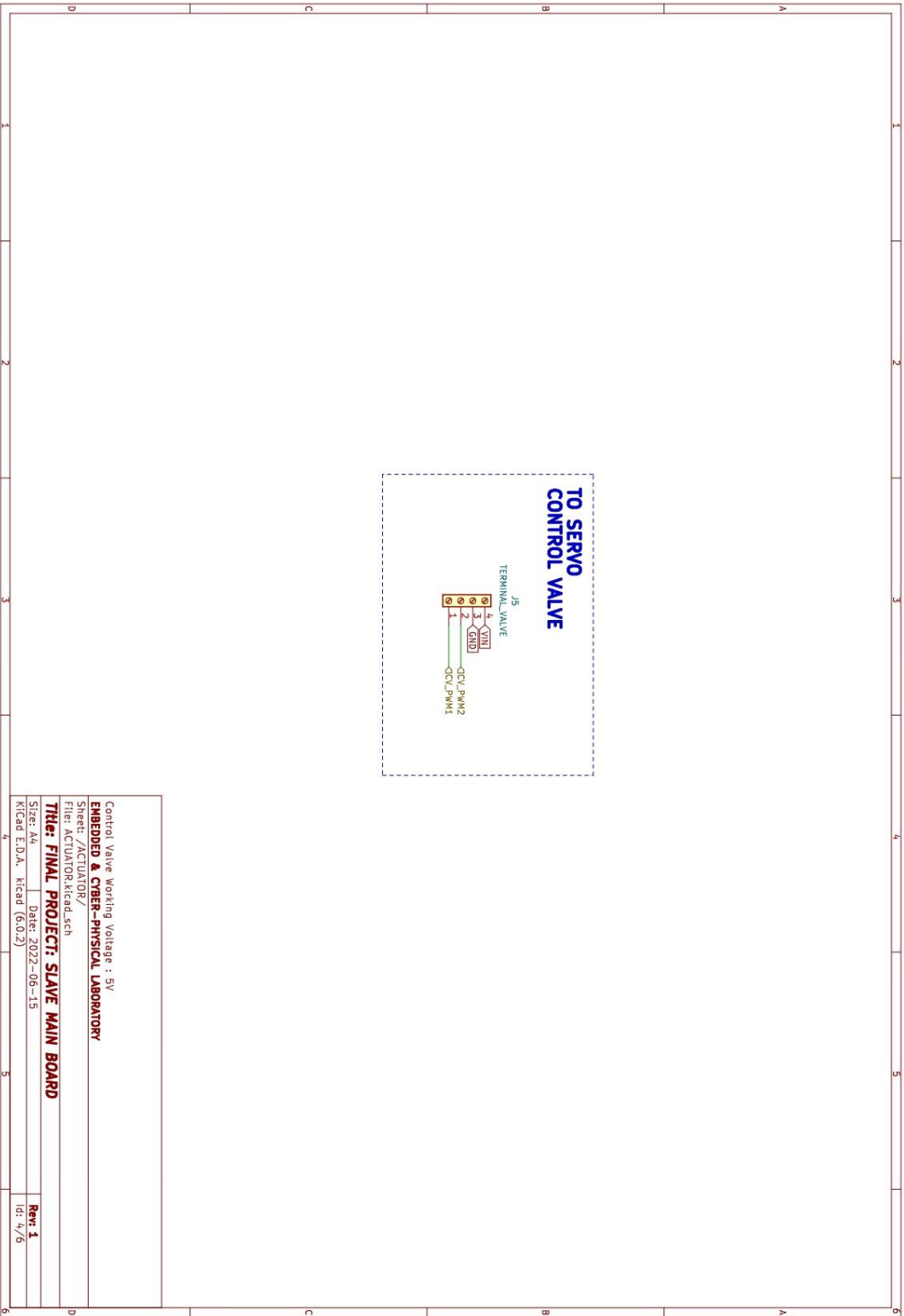


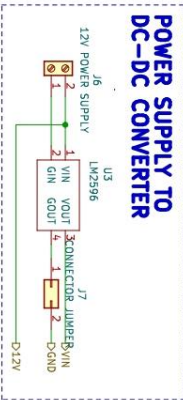




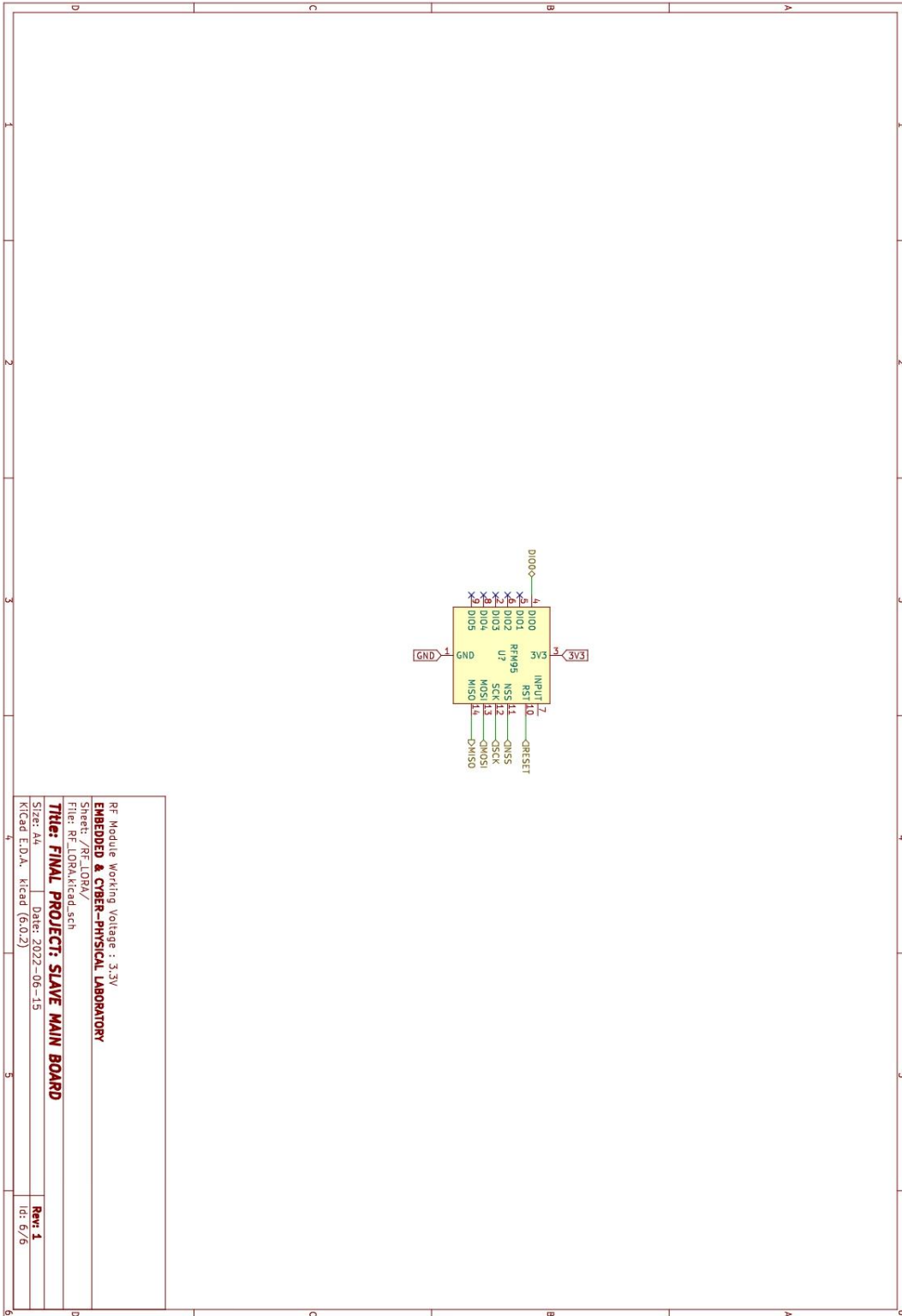
Sensor Working Voltage : 3.3V
EMBEDDED & CYBER-PHYSICAL LABORATORY
 Sheet /I2C_SENSOR/
 File: I2C_INPUT.ticed.sch
THIS IS FINAL PROJECT: SLAVE MAIN BOARD
 Size: A4 Date: 2022-06-15
 KICad E.D.A. Kicad (6.0.2)

Rev: 1
 Id: 3/6





EMBEDDED & CYBER-PHYSICAL LABORATORY	
Sheet: /POWER/	
File: POWER.kicad.sch	
Title: FINAL PROJECT: SLAVE MAIN BOARD	
Size: A4	Date: 2022-06-15
KiCad E.D.A. Kicad (6.0.2)	Rev: 1
	Id: 5/6



PE Module Working Voltage : 3.3V
EMBEDDED & CYBER-PHYSICAL LABORATORY
Schematics/PE-LORA/
Files/LORA/kicad.sch
Title: FINAL PROJECT: SLAVE MAIN BOARD
Size: A4 Date: 2022-06-15
Kicad E.D.A. Kicad (6.0.2)

Rev: 1
Id: 6/6

Lampiran B. Data Input Output Rancangan Neural network

B1. Data Input Output Pengendali Level Campuran

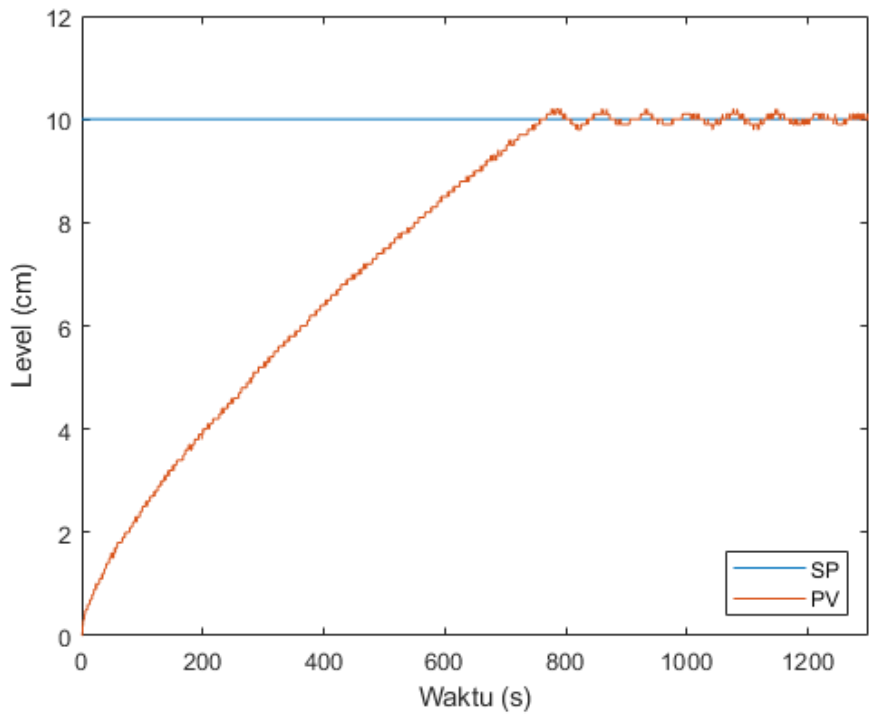
Campuran										
Input								Output		
x1	x2	x3	x4	x5	x6	x7	x8	kp	ki	kd
1	0	0	0	0	1	0	0	45	5	15
1	0	0	0	0	0	1	0	45	5	25
1	0	0	0	0	0	0	1	45	5	25
0	1	0	0	0	1	0	0	65	20	3
0	1	0	0	0	0	1	0	65	20	10
0	1	0	0	0	0	0	1	65	20	10
0	0	1	0	0	1	0	0	95	25	15
0	0	1	0	0	0	1	0	95	25	25
0	0	1	0	0	0	0	1	95	25	25
0	0	0	1	0	1	0	0	65	20	3
0	0	0	1	0	0	1	0	65	20	10
0	0	0	1	0	0	0	1	65	20	10
0	0	0	0	1	1	0	0	95	25	15
0	0	0	0	1	0	1	0	95	25	25
0	0	0	0	1	0	0	1	95	25	25

B2. Data Input Output Pengendali Level Minyak

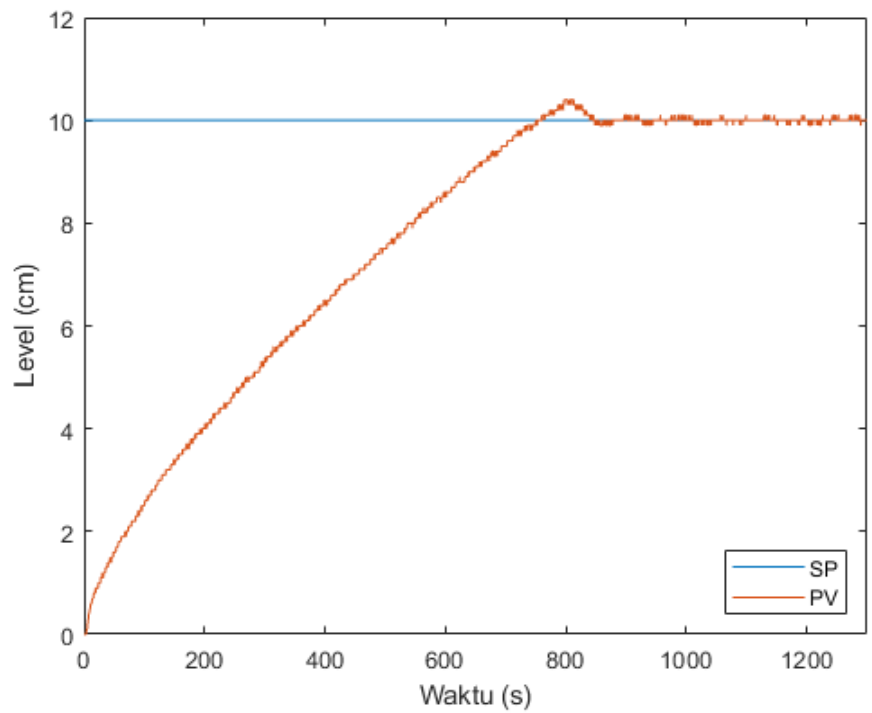
Minyak										
Input								Output		
x1	x2	x3	x4	x5	x6	x7	x8	kp	ki	kd
1	0	0	0	0	1	0	0	18	4.5	20
1	0	0	0	0	0	1	0	18	4.5	30
1	0	0	0	0	0	0	1	18	4.5	30
0	1	0	0	0	1	0	0	35	6	30
0	1	0	0	0	0	1	0	35	6	50
0	1	0	0	0	0	0	1	35	6	50
0	0	1	0	0	1	0	0	40	15	20
0	0	1	0	0	0	1	0	40	15	50
0	0	1	0	0	0	0	1	40	15	50
0	0	0	1	0	1	0	0	35	6	30
0	0	0	1	0	0	1	0	35	6	50
0	0	0	1	0	0	0	1	35	6	50
0	0	0	0	1	1	0	0	40	15	20
0	0	0	0	1	0	1	0	40	15	50
0	0	0	0	1	0	0	1	40	15	50

Lampiran C. Respon Dinamik Sistem Pengendalian PID Pada Level Campuran

C1. Variasi $K_p = 65$, $K_i = 20$, $K_d = 3$

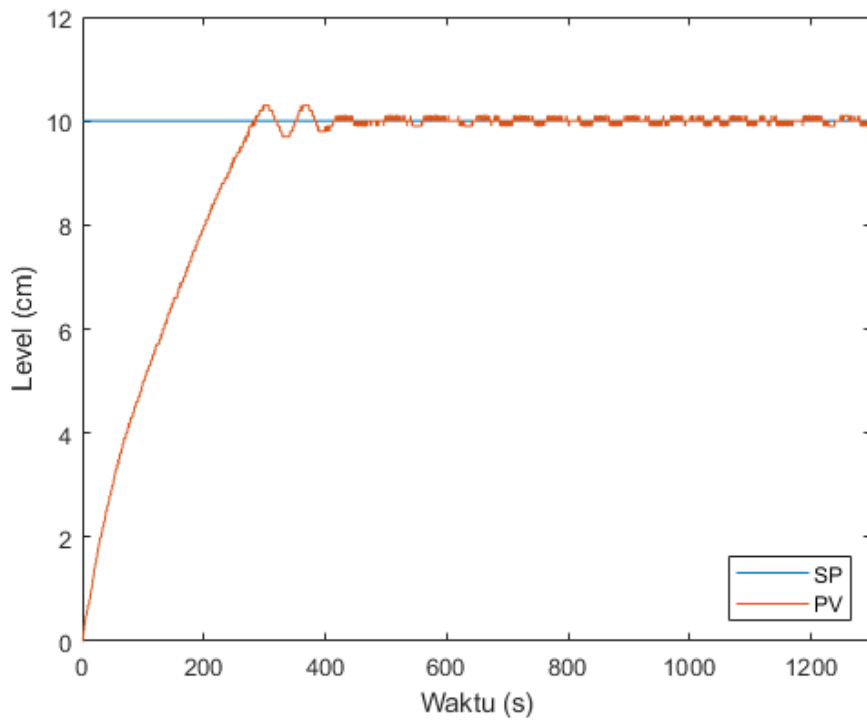


C2. Variasi $K_p = 95$, $K_i = 10$, $K_d = 15$

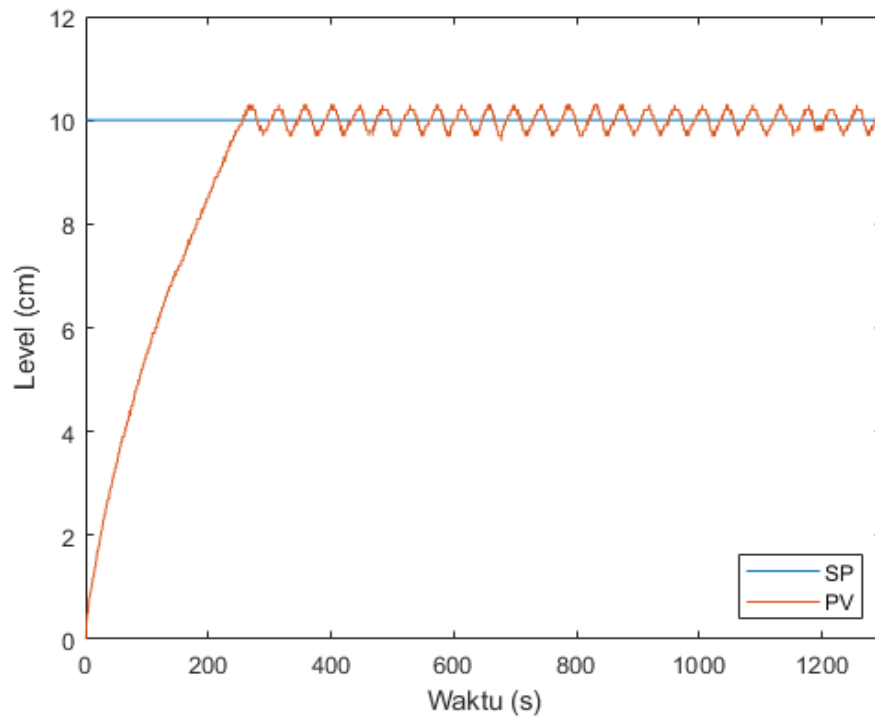


Lampiran D. Respon Dinamik Sistem Pengendalian PID Pada Level Minyak

D1. Variasi $K_p = 24$, $K_i = 15$, $K_d = 30$



D2. Variasi $K_p = 35$, $K_i = 15$, $K_d = 10$



Halaman ini sengaja dikosongkan

BIODATA PENULIS



Sulthan Ariq Prananda lahir di Kota Batam, 1 Juni 2000 adalah anak pertama dari tiga bersaudara. Penulis mengawali jenjang pendidikan formal mulai dari TK Al-Azhar Sekupang, kemudian SMPN 3 Batam, dan SMAN 1 Batam. Pada tahun 2018 penulis melanjutkan pendidikan pada jenjang sarjana di Departemen Teknik Fisika Institut Teknologi Sepuluh Nopember. Semasa kuliah penulis aktif sebagai asisten laboratorium dengan bidang penelitian pengembangan *Internet of Things*. Penulis juga tergabung dalam tim riset mobil hemat energi sebagai kepala divisi elektrikal dan sistem propulsi serta aktif dalam mengikuti kompetisi nasional. Penulis juga pernah mengikuti program Bangkit Academy pada 2021 sebagai *Mobile Developer* dan melakukan *internship* di PT. Ruang Cipta Teknologi sebagai *Embedded System Engineer*. Penulis dapat dihubungi melalui email sulthanariqq@gmail.com.