



TUGAS AKHIR - TE 141599

**PERANCANGAN JARINGAN KOMUNIKASI MODBUS
UNTUK SISTEM PENGATURAN PROSES PADA TANGKI
CPO**

Eko Arisana Ismawan
NRP 2214 105 070

Dosen Pembimbing
Ir. Ali Fatoni, M. T.
Imam Arifin, S.T, M. T.

PROGRAM STUDI TEKNIK SISTEM PENGATURAN
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE 141599

***DESIGN OF MODBUS COMMUNICATION NETWORK TO
PROCESS CONTROL SYSTEM IN CPO STORAGE***

Eko Arisana Ismawan
NRP 2214 105 070

Supervisor

Ir. Ali Fatoni, M. T.
Imam Arifin, S.T, M. T.

CONTROL SYSTEM ENGINEERING
Electrical Engineering Department
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul

Perancangan Jaringan Komunikasi Modbus untuk Sistem Pengaturan Proses pada Tangki CPO

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2016



Eko Arisana Ismawan
NRP 2214 105 070

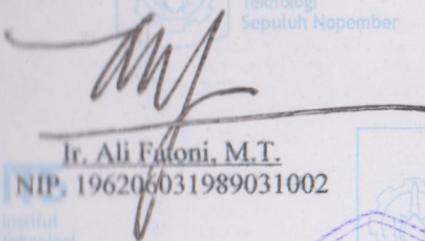
**PERANCANGAN JARINGAN KOMUNIKASI MODBUS UNTUK
SISTEM PENGATURAN PROSES PADA TANGKI CPO**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I,



Ir. Ali Fatoni, M.T.

NIP. 196206031989031002

Dosen Pembimbing II,



Imam Arifin, S.T., M.T.

NIP. 197302222002121001



PERANCANGAN JARINGAN KOMUNIKASI MODBUS PADA SISTEM PENGATURAN PROSES UNTUK TANGKI CPO

Nama : Eko Arisana Ismawan
NRP : 2214 105 070
Dosen Pembimbing I : Ir. Ali Fatoni, M.T.
NIP : 196206031989031002
Dosen Pembimbing II : Imam Arifin, S.T., M.T.
NIP : 197302222002121001

ABSTRAK

Pada umumnya, sebuah jaringan komunikasi di industri memerlukan sebuah sistem kerja yang *real-time*. Akan tetapi, dalam merancang sebuah jaringan untuk sistem pengaturan pada tangki CPO memiliki kendala yaitu adanya *delay* dan *packet loss* dari data yang dikirimkan. Adapun kriteria sistem komunikasi *real-time* yang harus dipenuhi salah satunya seperti batasan *delay* yang sesuai dengan standar IEC 61784-2 yaitu kurang dari 100 ms. Aplikasi Modbus dapat diterapkan dalam membangun sebuah jaringan untuk memenuhi kebutuhan komunikasi. Investigasi kualitas jaringan seperti *delay*, kecepatan data (*bandwidth*), *jitter*, dan *throughput* dapat dilakukan untuk mengetahui performansi dari jaringan yang dirancang. Hasil pengujian didapatkan bahwa performa dari jaringan komunikasi pada tangki CPO masih dalam keadaan baik dengan *delay* yang masih memenuhi standar yaitu sebesar 40 ms dengan kecepatan data yang dikirim sebesar 320 kbits/detik.

Kata Kunci : Tangki CPO, Jaringan Komunikasi, Modbus, Performa Jaringan.

**DESIGN MODBUS COMMUNICATION NETWORK TO PROCESS
CONTROL SYSTEM IN CPO STORAGE**

Author : Eko Arisana Ismawan
ID Number : 2214 105 070
Supervisor I : Ir. Ali Fatoni, M.T.
ID Number : 196206031989031002
Supervisor II : Imam Arifin, S.T., M.T.
ID Number : 197302222002121001

ABSTRACT

Generally, a communication network in the industry require a working system real-time. However, in designing a network for system settings on CPO tank has a constraint that is the delay and packet loss of data transmitted. The criteria for real-time communication system that must be met one such limitation delay in accordance with IEC 61784-2 standard that is less than 100 ms. Modbus application can be applied in building a network to meet the communication needs. Investigation of network quality such as delay, data rate (bandwidth), jitter, and throughput can be done to determine the performance of the network designed. The test results showed that the performance of the communication network on the tank CPO still in good condition with a delay that still meet the standards is 40 ms with a speed of data sent is 320 kbits / sec.

Keywords : CPO Storage, Communication Network, Modbus, Network Performance

DAFTAR ISI

HALAMAN

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN.....	v
LEMBAR PENGESAHAN	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Metodologi.....	2
1.6 Sistematika Penulisan	3
1.7 Relevansi.....	4
BAB II JARINGAN KOMUNIKASI MODBUS PADA SISTEM PENGATURAN PROSES	5
2.1 Tangki CPO	5
2.2 Sistem Pengaturan <i>Level</i> pada Tangki CPO.....	8
2.2.1 Pompa Tipe <i>Gear</i>	9
2.2.2 Sensor <i>Level</i> Radar	10
2.3 Sistem Pengaturan Temperatur pada Tangki CPO.....	11
2.3.1 Sensor <i>Resistance Temperature Detector</i> (RTD)	11
2.4 Arsitektur SCADA pada Tangki CPO	12
2.5 <i>Local Area Network</i> (LAN)	14
2.6 Topologi Jaringan	15
2.7 Protokol Jaringan di Industri	17
2.8 Performansi Jaringan	24
BAB III PERANCANGAN SISTEM	27
3.1 Pemodelan Sistem Pengaturan <i>Level</i>	27

3.1.1	Pemodelan <i>Gain</i> Pompa	28
3.1.2	Pemodelan Waktu Tunda karena Pipa.....	29
3.1.3	Pemodelan Tangki CPO	29
3.1.4	Pemodelan <i>Gain</i> Sensor	30
3.1.5	Perancangan Kontroler PI	31
3.2	Pemodelan Sistem Pengaturan Temperatur	33
3.2.1	Pemodelan Sensor RTD	36
3.2.2	Perancangan Kontroler PI	37
3.3	Perancangan Protokol Jaringan Modbus.....	38
3.3.1	Perancangan Komunikasi Kontroler dengan <i>Server</i> untuk Pengaturan <i>Level</i>	39
3.3.2	Komunikasi Kontroler dengan <i>Server</i> untuk Pengaturan Temperatur	41
3.3.3	Perancangan Komunikasi <i>Server</i> dengan <i>Client</i>	42
BAB IV	PENGUJIAN DAN ANALISA	45
4.1	Pengujian Kontroler PI Sistem Pengaturan <i>Level</i>	45
4.2	Pengujian Kontroler PI Sistem Pengaturan Temperatur	46
4.3	Pengujian Protokol Jaringan Modbus	47
4.4	Pengujian Performa Jaringan Modbus	49
4.4.1	Pengaruh Variasi <i>Message Size</i>	49
4.4.2	Pengaruh Variasi Jumlah Paket Data	50
4.4.3	Respon Sistem Pengaturan <i>Level</i>	53
4.4.4	Respon Sistem Pengaturan Temperatur.....	55
BAB V	PENUTUP	57
5.1	Kesimpulan.....	57
DAFTAR PUSTAKA		59

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	Tangki CPO	5
Gambar 2.2	Tangki Tipe <i>Fixed Roof Tank</i>	6
Gambar 2.3	Sistem Pengaturan <i>Level</i> Tangki CPO	9
Gambar 2.4	Pompa Tipe <i>Gear</i> Eksternal.....	10
Gambar 2.5	Sensor Radar	10
Gambar 2.6	Sistem Pengaturan Temperatur Tangki CPO	12
Gambar 2.7	Sensor RTD	12
Gambar 2.8	Arsitektur SCADA pada Tangki CPO	13
Gambar 2.9	Model Hubungan <i>Server-Client</i>	14
Gambar 2.10	Arsitektur Topologi <i>Bus</i>	15
Gambar 2.11	Arsitektur Topologi <i>Star</i>	16
Gambar 2.12	Arsitektur Topologi <i>Ring</i>	16
Gambar 2.13	Siklus Pengiriman <i>Query-Response</i>	17
Gambar 2.14	Diagram Alur Komunikasi TCP	21
Gambar 2.15	Paket Data Modbus TCP	22
Gambar 2.16	Mode Komunikasi <i>Full Duplex</i>	23
Gambar 2.17	Diagram Waktu Pengiriman Data.....	25
Gambar 3.1	Parameter-parameter <i>Level</i> Tangki CPO	27
Gambar 3.2	Diagram Blok Sistem Pengaturan <i>Level</i> pada Tangki CPO	28
Gambar 3.3	Diagram Blok Sistem dengan Kontroler PI	32
Gambar 3.4	Parameter-Parameter Temperatur Tangki CPO	33
Gambar 3.5	Diagram Blok Sistem Pengaturan Temperatur pada Tangki CPO	34
Gambar 3.6	Diagram Blok Sistem Pengaturan Temperatur dengan Kontroler PI.....	37
Gambar 3.7	Rancangan Arsitektur Jaringan Komunikasi.....	39
Gambar 3.8	HMI Pengaturan <i>Level</i> pada Tangki CPO	40
Gambar 3.9	HMI Pengaturan Temperatur pada Tangki CPO.....	42
Gambar 3.10	HMI <i>Server</i> pada Tangki CPO.....	42
Gambar 4.1	Diagram Blok Sistem Pengaturan <i>Level</i> dengan Kontroler PI	45
Gambar 4.2	Respon Sistem Pengaturan <i>Level</i> dengan Kontroler PI.....	45

Gambar 4.3	Diagram Blok Sistem Pengaturan Temperatur dengan Kontroler PI	46
Gambar 4.4	Respon Sistem Pengaturan Temperatur dengan Kontroler PI.....	46
Gambar 4.5	Status Komunikasi Modbus pada Sistem Pengaturan <i>Level</i>	47
Gambar 4.6	Status Komunikasi Modbus pada Sistem Pengaturan Temperatur	48
Gambar 4.7	Pengaruh Waktu Tunda Jaringan pada Pengaturan <i>Level</i>	49
Gambar 4.8	Pengaruh Waktu Tunda pada Pengaturan Temperatur	50
Gambar 4.9	Blok Konfigurasi Pengiriman Paket Data	50
Gambar 4.10	Perbandingan Jumlah Paket data terhadap <i>Delay</i>	51
Gambar 4.11	Perbandingan Jumlah Paket Data dengan <i>Bit Rate</i>	52
Gambar 4.12	Perbandingan Jumlah Paket Data dengan <i>Throughput</i>	52
Gambar 4.13	<i>Delay</i> Pengiriman Data untuk Pengaturan <i>Level</i>	53
Gambar 4.14	Kecepatan Pengiriman Data untuk Pengaturan <i>Level</i>	54
Gambar 4.15	Pengaruh Waktu Tunda 40 milidetik terhadap Kontroler PI pada Pengaturan <i>Level</i>	54
Gambar 4.16	<i>Delay</i> Pengiriman Data untuk Pengaturan Temperatur	55
Gambar 4.17	Kecepatan Pengiriman Data untuk Pengaturan Temperatur	56
Gambar 4.18	Pengaruh Waktu Tunda 40 milidetik terhadap Kontroler PI pada Pengaturan Temperatur	56

DAFTAR TABEL

HALAMAN

Tabel 2.1	<i>Nozzle</i>	7
Tabel 2.2	Pengalaman Modbus dan <i>Function Codes</i>	18
Tabel 2.3	<i>Exception Code</i> dalam <i>Exception Response</i>	20
Tabel 3.1	Parameter Sistem Pengaturan <i>Level</i> pada Tangki CPO	28
Tabel 3.2	Parameter-Parameter Sistem Pengaturan Temperatur Tangki CPO.....	34
Tabel 3.3	Parameter Sistem yang Diketahui Nilainya	36
Tabel 3.4	<i>Setting</i> Parameter Modbus untuk Pengaturan <i>Level</i> ..	39
Tabel 3.5	<i>Setting</i> Parameter Modbus untuk Pengaturan Temperatur.....	41
Tabel 3.6	<i>Setting</i> Parameter Modbus untuk Komputer <i>Server-Client</i>	43
Tabel 4.1	Pembacaan <i>Register</i>	48
Tabel 4.2	Performa Jaringan	49
Tabel 4.3	Variasi Jumlah Pengiriman Paket Data.....	51
Tabel 4.4	Performa Jaringan Sistem Pengaturan <i>Level</i>	53
Tabel 4.5	Performa Jaringan Sistem Pengaturan Temperatur...	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tangki timbun merupakan tempat penyimpanan untuk keperluan penyediaan *Crude Palm Oil* (CPO). Tangki tersebut dilengkapi dengan sensor temperatur untuk mendeteksi temperatur pada CPO. Tangki timbun juga dilengkapi dengan sistem pemanas *boiler* agar kualitas CPO terjaga dalam temperatur antara 50-55°C. Instrumen lain yang digunakan yaitu sensor *level* dan *flowmeter* yang digunakan untuk membaca *level* CPO dan mengetahui masukan dan keluaran aliran CPO. Semua instrumen tersebut diintegrasikan ke dalam sebuah sistem dalam satu pusat kendali.

Penggunaan sistem pengendalian terdistribusi telah menjadi tren dalam industri proses dan otomasi[1]. Teknologi tersebut menyediakan fasilitas untuk memenuhi kebutuhan dalam monitoring dan pengendalian untuk sistem pengaturan yang ada pada tangki timbun. Fasilitas Supervisory Control and Data Acquisition (SCADA) banyak digunakan dalam sistem pengaturan proses di industri. Selama beberapa tahun SCADA menyediakan solusi bagi *level* supervisory di industri untuk mengkoordinasikan berbagai instrumen dan pengawasan terhadap peralatan sensor dan proses operasi lainnya[2]. Instrumen-instrumen tersebut didukung dengan komunikasi yang handal untuk menunjang pengendalian terhadap proses yang ada dalam tangki timbun. Berbagai jenis jaringan telah diperkenalkan untuk sistem pengendalian seperti Ethernet Modbus, ControlNet, DeviceNet, dan sebagainya[11] Pertimbangan untuk mengetahui karakteristik protokol komunikasi diperlukan dalam menentukan jaringan utama untuk menghubungkan antar instrumen pada SCADA.

Pada penelitian [3] *wireless* LAN digunakan untuk komunikasi industri yang *real-time* karena memberikan keuntungan dari jaringan nirkabel dan dari segi ketepatan waktu dan kehandalan. Permasalahan yang muncul dalam komunikasi yaitu waktu yang dibutuhkan sebuah data dikirim agar sampai diterima oleh *receiver* dan tingkat keberhasilan pengiriman sebuah data dari satu komputer ke komputer lainnya karena dalam sebuah jaringan dimungkinkan sebuah data dapat hilang akibat adanya gangguan saat transmisi melalui suatu jaringan.

Dengan demikian, sebuah jaringan dengan protokol Modbus dibangun agar mampu mengendalikan sistem pengaturan *level* dan

temperatur yang terdapat pada tangki CPO dan menyelesaikan permasalahan yang muncul dalam jaringan komunikasi. Investigasi kualitas jaringan seperti *delay*, kecepatan data (*bandwidth*), *jitter*, dan *throughput* dapat dilakukan untuk mengetahui performansi dari jaringan yang dirancang.

1.2 Permasalahan

Penggunaan jaringan komunikasi pada tangki CPO untuk mendukung integrasi antar instrumen dalam melakukan pengendalian proses yang terhubung dalam suatu pusat kendali. Merancang sebuah jaringan untuk sistem pengaturan *level* dan temperatur pada tangki CPO muncul kendala yaitu adanya *delay* dan *packet loss* dari data yang dikirimkan. Adapun jaringan komunikasi yang dirancang harus memenuhi kriteria sistem yang *real-time* salah satunya seperti batasan *delay* sesuai dengan standar IEC 61784-2 yaitu kurang dari 100 ms.

1.3 Batasan Masalah

Pada penelitian ini membahas mengenai perancangan jaringan komunikasi menggunakan protokol Modbus untuk sistem pengaturan *level* dan temperatur. Untuk mengetahui kualitas jaringan tersebut dilakukan analisa terhadap parameter performa jaringan yang dibangun seperti *delay*, kecepatan data (*bandwidth*) dan *throughput*

1.4 Tujuan Penelitian

Pengendalian sebuah sistem pengaturan proses dalam industri besar membutuhkan kehandalan dalam segi peralatan maupun pengoperasian untuk mendukung performa sistem yang *real-time*. Sebuah sistem terintegrasi yang handal diperlukan agar dapat mengawasi dan mengatur jalannya sistem pengaturan proses yang terdapat pada tangki CPO. Tujuan penelitian ini yaitu membangun sebuah jaringan komunikasi untuk mengendalikan sistem pengaturan proses yang terdapat pada tangki CPO.

1.5 Metodologi

Pada topik penelitian ini dilakukan melalui beberapa tahapan, diantaranya studi literatur, membuat rancangan jaringan komunikasi yang dibangun meliputi menyusun komponen-komponen utama dan pendukung jaringan, merancang topologi jaringan yang digunakan, membuat simulasi komunikasi menggunakan protokol Modbus, pengujian

dan analisa performa jaringan pada protokol Modbus. Pengerjaan tahap awal hingga akhir digunakan sebagai bahan penyusunan buku laporan penelitian.

1.6 Sistematika Penulisan

Penulisan laporan ini dibagi dalam beberapa bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Pada tahapan ini berisi tentang penjabaran latar belakang penelitian. Berdasarkan latar belakang diperoleh permasalahan yang terjadi pada perancangan jaringan komunikasi untuk sistem pengaturan Tangki CPO. Pada bab ini juga membatasi masalah, tujuan, metodologi, dan sistematika penulisan yang dibahas dalam penelitian ini.

Bab II Jaringan Komunikasi Modbus pada Sistem Pengaturan Proses

Teori dasar mengenai arsitektur jaringan komunikasi, topologi jaringan, protokol komunikasi, performa jaringan, dan sistem pengaturan proses dibahas pada bab ini. Penjelasan tersebut diperoleh dari beberapa sumber meliputi buku, *paper*, dan *datasheet*.

Bab III Perancangan Sistem

Perancangan sistem jaringan komunikasi, mulai dari arsitektur sistem, perancangan komunikasi antar peralatan dan pengaplikasian protokol Modbus pada jaringan komunikasi yang dibangun, dijabarkan pada bab ini.

Bab IV Pengujian dan Analisis

Pada tahap ini dibahas tentang hasil pengujian yang dilakukan dengan variasi jumlah pengiriman paket data dan variasi besar paket data yang dikirim terhadap performa jaringan.

Bab V Penutup

Setelah pengujian dan analisa selesai dilakukan, disajikan beberapa kesimpulan dan juga disertakan saran mengenai kekurangan dari penelitian ini agar bisa dikembangkan berikutnya.

1.7 Relevansi

Penelitian ini diharapkan dapat bermanfaat sebagai pengetahuan dan gambaran untuk mahasiswa mengenai jaringan komunikasi dan dapat merancang sebuah jaringan untuk dapat diaplikasikan sesuai dengan kebutuhan yang ada di dunia industri.

BAB II

JARINGAN KOMUNIKASI MODBUS PADA SISTEM PENGATURAN PROSES

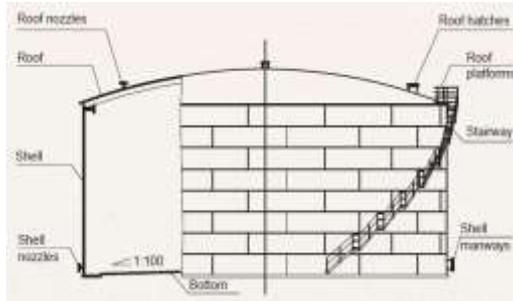
2.1 Tangki CPO

Tangki ini dibangun untuk memenuhi kebutuhan bahan baku CPO (*Crude Palm Oil*) dan PKO (*Palm Kernel Oil*) di kawasan Industri Sei Mangkei. Tangki yang digunakan ada 2 macam yaitu tangki *buffer* dan tangki timbun. Tangki *buffer* untuk menampung minyak sementara sebelum disalurkan ke tangki timbun sedangkan tangki timbun digunakan untuk menimbun minyak yang berasal dari produksi untuk diolah dan didistribusikan menuju konsumen[4].



Gambar 2.1 Tangki CPO [4]

Dalam industri kelapa sawit mempunyai spesifikasi dalam menentukan jenis tangki untuk penimbunan. Jenis tangki *Fixed Roof Tank* digunakan khusus untuk penyimpanan dan pengolahan kelapa sawit. Tangki jenis ini adalah tangki silinder dengan konfigurasi atap yang bersatu dengan dinding *shell* tangki. Bentuk *roof* tangki dapat berbentuk *cone* (kerucut) atau *dome* (kubah) yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 Tangki Tipe *Fixed Roof Tank* [4]

2.1.1 Spesifikasi Tangki Buffer [4][5]

Dalam tangki timbun CPO yang digunakan memiliki kriteria spesifikasi tertentu untuk memenuhi standar operasional. Kriteria tersebut diuraikan beserta pada Tabel 2.1 sebagai berikut.

- a. Bentuk : *Vertikal tank with fix roof and coiled*
- b. Fungsi : Menyimpan CPO
- c. Kapasitas : 5.000 Ton
- d. Jumlah : 10 unit
- e. Dimensi : Diameter x Tinggi = 21.350 mm x 14.662 mm
- f. Detail : *CPO-5000-TK01-12-003_0 General Assy Drawing For CPO Cap 5000m³-Model*
- g. *Top main way* : Φ 600 mm
- h. *Side main way* : Φ 600 mm
- i. Material : Carbon steel
- j. Tingkat 1 s/d 3 : 10 mm
- k. Tingkat 4 s/d 6 : 8 mm
- l. Tingkat 7 s/d 8 : 6 mm
- m. *Roof* : *Plate* : 4,5 mm
- n. *Structure* : Besi siku : 75 x75 x6 mm
- o. *Inner Ring* : UNP : 150 x 75 mm

Tabel 2.1 Nozzle

<i>Users</i>	<i>Size (mm)</i>	<i>Qty</i>	<i>Panjang (mm)</i>	<i>Flange</i>	<i>Material</i>
<i>Oil inlet</i>	100	1	150	<i>JIS 16K</i>	<i>Carbon steel Sch 40</i>
<i>Oil outlet</i>	150	1	150	<i>JIS 16K</i>	<i>Carbon steel Sch 40</i>
<i>Drain</i>	100	1	150	<i>JIS 16K</i>	<i>Carbon steel Sch 40</i>
<i>Vent</i>	150	1	150	<i>JIS 16K</i>	<i>Carbon steel Sch 40</i>
<i>Steam inlet</i>	50	2	150	<i>JIS 16K</i>	<i>Seamless API 5L Gr B Sch 40</i>
<i>Cond.outlet</i>	50	2	150	<i>JIS 16K</i>	<i>Seamless API 5L Gr B Sch 40</i>
<i>Oil recycle</i>	80	1	150	<i>JIS 16K</i>	<i>Carbon steel Sch 40</i>

2.1.2 Loading CPO dan PKO

Fasilitas *loading* CPO dan PKO ini terdiri dari peralatan jembatan timbang, *buffer tank* dan pompa *loading*. *Loading* CPO dan PKO akan dilakukan dengan 3 cara, yaitu *transport* menggunakan truk tangki, kereta api dari *dryport* dan pipanisasi langsung dari pabrik. Untuk *loading* dari *dryport* dan pipanisasi dari *dryport system* pengukuran CPO dan PKO yang di *loading* dilakukan dengan cara pengukuran *volumetric* menggunakan *flowmeter*, sedangkan pengukuran CPO dan PKO yang di *loading* menggunakan truk tangki dilakukan dengan timbangan. Cara *loading* menggunakan pipanisasi dilakukan dengan memompakan langsung CPO dari *dryport* dan dari pabrik ke *buffer tank* masing – masing untuk kemudian di pompakan ke tangki masing – masing untuk ditimbin.

Loading menggunakan truk tangki dilakukan ketika truk tangki masuk kedalam kompleks tangki timbun, truk langsung menuju ke truk dispenser untuk melakukan pengurusan surat dan pengambilan sampel CPO dan PKO. Sampel yang diambil kemudian di cek kadar FFA, air dan Dobinya. Jika kadar FFA dan kadar air maupun dobinya tidak memenuhi syarat maka akan dilakukan negoisisasi menurut aturan yang berlaku. Jika kualitas CPO dan PKO memenuhi syarat truk tangki langsung menuju penimbangan untuk diketahui berat total kendaraannya, saat penimbangan supir diharuskan keluar dari dalam truk. Setelah dicatatkan

beratnya truk langsung menuju ke *buffer tank* untuk melakukan pengeluaran CPO dan PKO yang dibawanya. Setelah pengeluaran CPO dan PKO selesai truk tangki kembali menuju timbangan untuk mengetahui berat kosongnya. Selisih antara berat keseluruhan dengan berat kosong merupakan berat CPO atau PKO yang dikeluarkan di *buffer tank*.

2.1.3 Penimbunan CPO atau PKO

Tangki timbun yang harus disediakan untuk memenuhi kapasitas timbun sebesar 40.000 Ton untuk CPO dan 6.000 Ton untuk PKO adalah 8 buah tangki dengan kapasitas 5.000 Ton untuk CPO dan 2 buah tangki 3.000 Ton untuk PKO. Tiap tangki akan dilengkapi dengan sistem monitoring digital dan dapat diatur pengeluaran dan stok didalamnya secara otomatis. Selain itu tiap tangki dilengkapi dengan pipa pemanas untuk memenuhi syarat bongkar muat sesuai dengan ketentuan yang berlaku. Temperatur dalam tangki diatur agar saat pembongkaran berkisar pada temperatur 50 – 55°C, akan tetapi pola pemanasannya hanya diperbolehkan meningkatkan temperatur 5°C per hari. Bongkar muat dari tangki dan menuju konsumen akan dilakukan secara bergiliran menggunakan tangki yang terlebih dahulu diisi atau sering disebut dengan sistem FIFO (*First in First Out*) sehingga kualitas CPO atau PKO akan selalu terjaga. Tangki timbun ini dibuat dari bahan logam *mild steel*.

2.1.4 Unloading CPO dan PKO

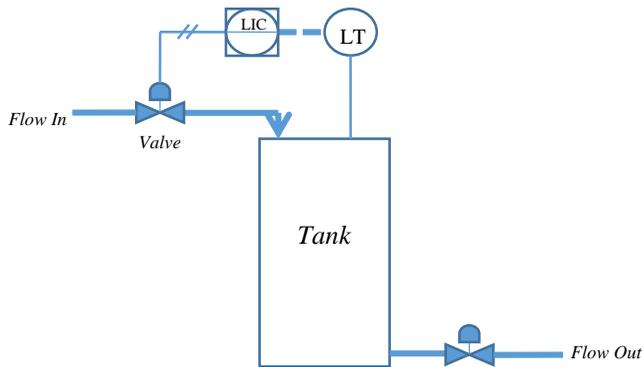
Fasilitas ini terdiri dari pemipaan, *motorized valve*, pompa dan *metering system*. Fasilitas ini menganut sistem FIFO (*first in frist out*) dimana CPO atau PKO yang ditimbun dalam tangki akan dialirkan kepada konsumen menurut tangki mana yang di isi terlebih dahulu. Pembukaan tangki mana yang akan dialirkan dapat dilakukan dengan kendali jarak jauh langsung dari *Central Control Room*. Setelah ditentukan tangki mana yang akan dikosongkan maka CPO datu PKO dapat dialirkan ke konsumen melalui *flowmeter* yang akan mencatat jumlah CPO yang dialirkan. Sebelu dialirkan, temperatur CPO harus diseragamkan agar mempermudah melakukan transfer dan bacaan *flowmeter* menjadi lebih akurat.

2.2 Sistem Pengaturan Level pada Tangki CPO

Pada tangki terdapat pengaturan *level* yang membutuhkan beberapa komponen untuk mendukung proses meliputi sensor *level*,

kontroler dan aktuator. Sensor *level* mendeteksi *level* cairan yang ada pada tangki. Sinyal sensor diolah *transmitter* agar dapat dibaca oleh kontroler yaitu membandingkan nilai *level* yang diinginkan dengan nilai aktual pembacaan sensor. Selisih antara kedua nilai tersebut dinamakan kesalahan.

Kontroler menggerakkan aktuator untuk mengurangi kesalahan. Aktuator dapat berupa pompa atau *valve* untuk mengalirkan cairan menuju tangki. Semua komponen tersebut beroperasi secara kontinu, sehingga pada waktu tertentu kesalahan antara nilai *level* yang diinginkan dengan nilai pembacaan sensor dapat hilang.

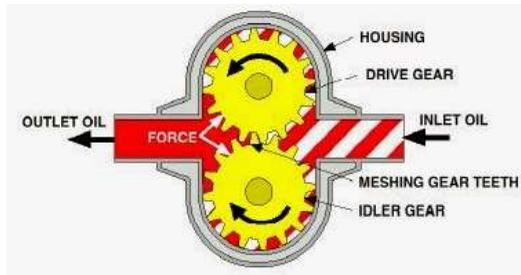


Gambar 2.3 Sistem Pengaturan *Level* Tangki CPO

Seperti yang diilustrasikan pada Gambar 2.3, terdapat tiga komponen utama yaitu LT (*Level Transmitter*), LIC (*Level Indicator Controller*), *valve*. LT sebagai sensor dan *transmitter*, LIC sebagai kontroler dan *valve* sebagai aktuator.

2.2.1 Pompa Tipe *Gear*

Pada Gambar 2.4 menunjukkan pompa dengan tipe *gear* eksternal yang bekerja dengan cara mengalirkan fluida melalui celah-celah antara gigi melalui *outlet*. Sifat pasangan roda gigi selalu memiliki titik kontak, sehingga dapat dimanfaatkan oleh mekanisme pompa tipe *gear* untuk mengalirkan fluida. Dengan kata lain, secara ideal fluida tidak akan masuk melalui titik kontak pasangan roda gigi tersebut.



Gambar 2.4 Pompa Tipe *Gear* Eksternal [4]

Jika jumlah gigi semakin sedikit, maka *volume* fluida yang dialirkan semakin besar karena rongga antara roda gigi dengan dinding semakin besar pula. Untuk meningkatkan *flowrate* dapat dilakukan dengan meningkatkan *rpm* dari roda gigi tersebut. Pompa jenis ini tidak memerlukan *valve* sehingga *loss* dapat berkurang.

2.2.2 Sensor Level Radar [4]

Instrumen *level non-contact radar* pada Gambar 2.5 mengukur jarak sensor/*transmitter* (yang berlokasi pada beberapa titik puncak) ke permukaan cairan dalam tangki. Instrumen *non-contact radar* menggunakan sebuah antena untuk memancarkan atau mengirim gelombang radio ke cairan yang ada di dalam tangki untuk diketahui *level*-nya. Perangkat yang digunakan yaitu sensor *level* ultrasonik dimana cara kerjanya dengan mengukur waktu gelombang berjalan untuk menentukan *level* cairan. Pengukuran menggunakan radar mempunyai dua teknik modulasi utama, yaitu radar pulsa dan teknik radar FCMW (*Frequency Modulated Continuous Wave*).



Gambar 2.5 Sensor Radar [4]

Teknik radar pulsa mengirim sinyal *microwave* yang memantulkan permukaan cairan dan kembali ke *gauge*. *Transmitter* mengukur waktu tunda antara yang ditransmisikan dan yang diterima sinyal *echo* dan mikroprosesor *on-board* menghitung jarak ke permukaan cairan menggunakan Persamaan 2.1.

$$\text{Jarak} = [\text{kecepatan cahaya} \cdot \text{waktu tunda}] / 2 \quad 2.1$$

Perbedaan yang mendasar antara instrumen *level* radar dan instrumen *level* ultrasonik adalah jenis gelombang yang digunakan. Radar menggunakan gelombang radio sebagai ganti gelombang suara yang digunakan oleh ultrasonik. Berikut dijelaskan tentang spesifikasi sensor radar yang digunakan dalam tangki CPO.

1. Jarak ukur hingga 80 meter.
2. Tingkat akurasi pengukuran hingga ± 3 mm.
3. Dapat digunakan dalam temperatur hingga 200°C.
4. Pengukuran handal dalam berbagai macam kondisi proses.

2.3 Sistem Pengaturan Temperatur pada Tangki CPO

Sistem pengaturan temperatur di industri secara umum meliputi sistem tangki, kontroler dan sensor temperatur yang digunakan pada sistem tersebut. Banyak industri yang sudah menerapkan metode pengendalian tertentu untuk sistem tersebut. Akan tetapi, banyak juga yang masih menerapkan kontroler *on/off* tanpa memperhatikan kondisi *plant*, sehingga pada bab ini dipilih untuk membahas sistem tangki dan sensor temperatur yang banyak digunakan di industri yaitu *Resistance Temperature Detector* (RTD).

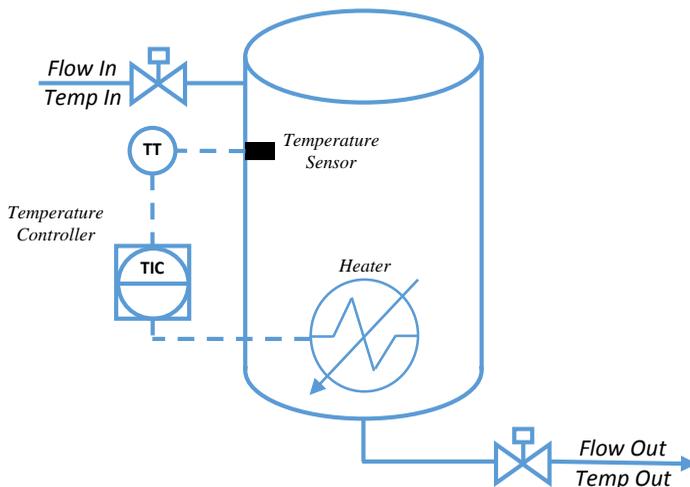
2.3.1 Sensor *Resistance Temperature Detector* (RTD)

Sensor RTD beroperasi berdasarkan prinsip hambatan listrik. Elemen yang biasanya digunakan untuk membuat RTD adalah platinum. Elemen tersebut dipilih karena mampu mengukur suhu dengan jangkauan yang luas, memiliki tingkat akurasi dan stabilitas yang tinggi.



Gambar 2.6 Sensor RTD [4]

RTD memiliki nilai hambatan yang berubah-ubah berdasarkan suhu yang terukur. Perubahan nilai hambatan RTD berbanding lurus terhadap suhu terukur olehnya. Suhu yang mampu terukur oleh sensor RTD bervariasi tergantung elemen penyusunnya. Misalnya untuk RTD yang elemen penyusunnya platinum, memiliki jangkauan temperatur antara -200°C sampai 800°C . Salah satu jenis RTD yang digunakan di industri seperti pada Gambar 2.6.



Gambar 2.7 Sistem Pengaturan Temperatur Tangki CPO

2.4 Arsitektur SCADA pada Tangki CPO

SCADA adalah sistem yang digunakan untuk mengawasi dan mengatur sebuah proses di industri. Istilah SCADA merujuk pada sistem terpusat untuk mengawasi dan mengatur beberapa *site* atau proses yang tersebar luas mulai dari satu *plant* sampai beberapa *plant*. Sebagian besar

aksi kontrol pada proses dilakukan secara otomatis oleh RTU atau PLC. Komponen dasar SCADA pada tangki timbun ditunjukkan pada Gambar 2.8.

a. *AFD (Active Field Device)*

Salah satu bagian dari SCADA yang merupakan sebuah alat untuk mengaktifkan dan menghubungkan sensor *level* dan temperatur yang terdapat pada tangki dengan *DP/PA Coupler*. AFD dirancang di ruang lingkup terbuka sehingga instrumen ini tahan terhadap cuaca yang ekstrim seperti panas, dingin, hujan dan badai.

b. *DP/PA Coupler*

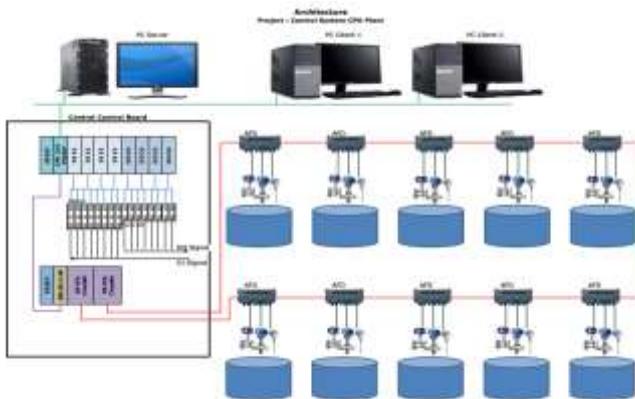
Instrumen ini berfungsi untuk menghubungkan *master* dengan AFD. *Profibus* digunakan sebagai komunikasi antara *master* dengan instrumen ini dan terhubung pula dengan AFD.

c. *Master Central Control (MCC)*

Sistem SCADA pada tangki CPO menggunakan PLC yang digunakan sebagai *server* untuk mengumpulkan data dari masing-masing sensor yang terhubung dengan AFD. *Server* terintegrasi dengan *DP/PA Coupler* dan panel pada ruang kontrol.

d. *PC Server dan PC Client (HMI)*

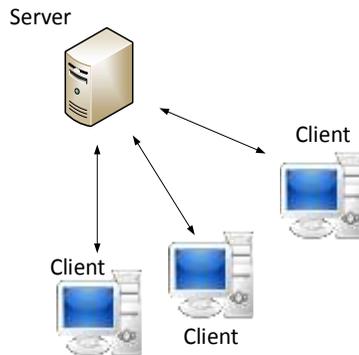
Instrumen ini dilengkapi dengan HMI yang digunakan untuk proses monitoring aliran, *level*, dan temperatur pada tangki CPO. Instrumen ini juga dapat menampilkan grafik data untuk masing-masing tangki dan alarm jika terjadi gangguan atau masalah pada tangki timbun.



Gambar 2.8 Arsitektur SCADA pada Tangki CPO

2.5 Local Area Network (LAN)

Jaringan memiliki beberapa aturan agar semua perangkat yang ada di industri dapat saling berkomunikasi [6]. Salah satunya adalah *Local Area Network* (LAN) yang dapat digunakan untuk menghubungkan antar peralatan (komputer) dalam lingkup skala industri. Penggunaan LAN sebagai jaringan komunikasi dalam industri karena mempunyai kapasitas kecepatan yang tinggi.



Gambar 2.9 Model Hubungan *Server-Client* [6]

Model hubungan *client/server* memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi kepada satu atau dua *dedicated file server* seperti pada Gambar 2.9. Sebuah *file server* menjadi pusat keseluruhan sistem yang memungkinkan untuk mengakses sumber daya dan menyediakan keamanan.

Workstation yang berdiri sendiri dapat mengambil sumber daya yang ada pada *file server*. Model *client/server* menyediakan mekanisme integrasi seluruh komponen yang ada di jaringan dan memungkinkan banyak pengguna secara bersama-sama memakai sumber daya pada *file server*. Prinsip kerja *client-server* sebagai berikut.

- a. *Client* mengirimkan data yang telah diberikan ke alamat *server* tujuan.
- b. Semua *server* dapat membaca data yang terdapat pada jaringan *bus* namun jika alamat tujuan data tidak cocok dengan *address* maka *server* akan mengabaikan data tersebut.

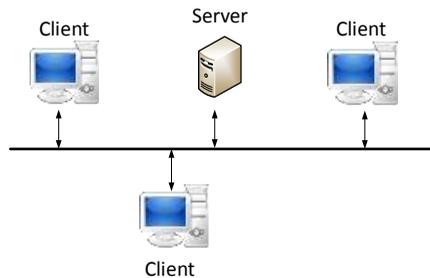
- c. *Server* akan membaca data dengan benar dan memeriksa setiap kesalahan (contohnya *frame size* dan *CRC error*).
- d. Jika alamat *server* tidak merespon, *master* mencoba untuk mengirimkan permintaan sebelum mengirim data untuk *server* selanjutnya.
- e. *Client* akan men-*cycle* semua permintaan data *bus*.

2.6 Topologi Jaringan

Membangun sebuah sistem komunikasi di industri perlu pertimbangan mengenai penggunaan topologi jaringan yang digunakan. Topologi merupakan suatu bentuk atau struktur jaringan yang menghubungkan antar komputer satu dengan yang lain menggunakan media berupa kabel maupun nirkabel. Topologi yang paling umum digunakan untuk jaringan di industri yaitu *star*, *bus*, dan *ring*.

2.6.1 Topologi Bus

Komunikasi menggunakan topologi ini yaitu antar perangkat (*client* dan *server*) terhubung langsung dalam satu jalur jaringan utama seperti pada Gambar 2.10. Pengiriman data dapat diketahui oleh perangkat lain namun data tersebut hanya akan sampai sesuai dengan alamat tujuannya.

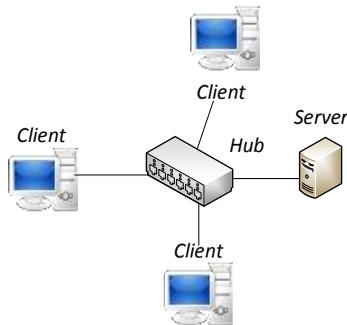


Gambar 2.10 Arsitektur Topologi *Bus* [6]

Keunggulan jenis topologi ini adalah pengembangan jaringan atau penambahan komputer dapat dilakukan namun dengan syarat tidak mengganggu komunikasi komputer lain. Kelemahannya bila terdapat gangguan di sepanjang kabel maka seluruh jaringan akan ikut mengalami gangguan.

2.6.2 Topologi Star

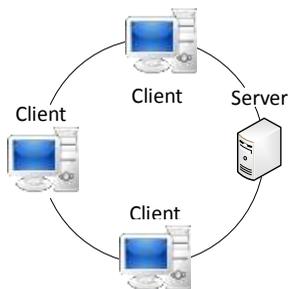
Konfigurasi jenis topologi ini memiliki pusat kontroler yang terhubung dengan semua komputer. Topologi ini mudah digunakan dalam lingkup jaringan skala kecil. Apabila terjadi kegagalan pada salah satu komputer dalam konfigurasi, topologi ini tidak mempengaruhi komputer yang lain. Sesuai pada Gambar 2.11, topologi *star* memiliki instrumen *hub* yang berada pusat jaringan. Topologi jenis ini dapat dengan mudah menambahkan komputer tambahan tanpa mengganggu jaringan.



Gambar 2.11 Arsitektur Topologi *Star* [6]

2.6.3 Topologi Ring

Seperti Gambar 2.12, setiap titik terkoneksi ke dua titik lainnya, membentuk jalur melingkar membentuk cincin. Pesan yang dikirim sekitar ring melewati setiap komputer. Jika salah satu komputer gagal dalam mengirim data maka seluruh jaringan berhenti kecuali mekanisme pemulihan tidak diterapkan.



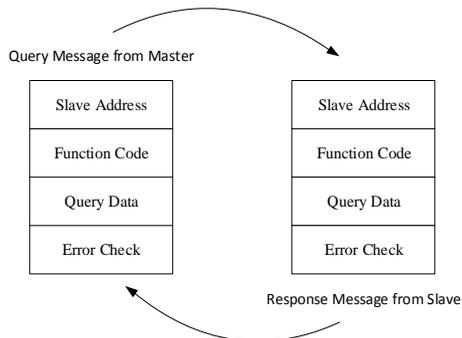
Gambar 2.12 Arsitektur Topologi *Ring* [6]

2.7 Protokol Jaringan di Industri

Protokol berperan mengatur peralatan di industri dapat berkomunikasi di mana dapat mengurangi waktu yang dibutuhkan dalam mengirimkan informasi, memastikan keamanan data, dan sinkronisasi waktu, serta respon waktu dari beberapa aplikasi yang *real-time* [7]. Protokol memastikan bahwa sistem dapat mengirimkan informasi handal tanpa adanya kesalahan dan aman. Beberapa protokol yang banyak digunakan dalam mengirimkan data seperti Modbus, Ethernet Modbus/TCP [8].

2.7.1 Modbus

Protokol jenis ini menerapkan aturan komunikasi data dengan dengan teknik *master-slave*. Komunikasi tersebut terdapat satu *master* dan beberapa *slave* yang membentuk sebuah jaringan. Seperti siklus pada Gambar 2.13, Modbus selalu diawali dengan *query* dari *master* dan *slave* yang memberikan respon dengan mengirimkan data sesuai perintah dari *master*.



Gambar 2.13 Siklus Pengiriman *Query-Response* [9]

Master hanya melakukan satu komunikasi dalam satu waktu. *Slave* hanya akan melakukan komunikasi jika ada perintah (*query*) dari *Master* dan tidak bisa melakukan komunikasi dengan *slave* yang lain[9]. Pada saat mengirimkan *query* ke *slave*, *master* menggunakan 2 mode pengalamatan, yaitu:

- Unicast mode* yaitu *master* mengirimkan *query* kepada satu *slave*. Setelah menerima dan memproses *query*, *slave* memberikan jawaban berupa respon kepada *master*.

- b. *Broadcast mode* yaitu *master* mengirimkan perintah (*query*) kepada semua *slave*. Mode pengalamatan ini *slave* tidak mengirimkan respon kepada *master*.

Protokol Modbus memiliki format pesan dalam bentuk *frame* bertujuan untuk mengirimkan data. *Frame* tersebut terdiri alamat tujuan, fungsi perintah, isi data, dan *error check*. Penjelasan masing-masing bagian pada *frame* Modbus dijelaskan sebagai berikut.

a. *Address Field*

Masing-masing *slave* harus mempunyai alamat yang berbeda dalam *range* 1 – 247 untuk pengalamatan individu. Alamat 0 digunakan untuk pengalamatan *broadcast*.

b. *Function Field*

Pada *frame* pesan ini berisi nomer kode fungsi (*function code*). Kode fungsi yang valid mempunyai *range* 1 – 255, di mana kode 1 – 127 untuk fungsi normal, sedangkan 128 – 255 untuk fungsi *exception response*. *Function code* berfungsi untuk memberitahu *slave* tentang perintah yang harus dikerjakan dan sebagai indikasi respon normal atau jenis *error* yang terjadi (*exception response*). Pada sistem komunikasi Modbus, jumlah *function code* yang didukung bervariasi tergantung kontroler dan peralatan *slave* yang digunakan. Beberapa kode fungsi berikut keterangannya ditunjukkan pada Tabel 2.2.

Tabel 2.2 Pengalamatan *Modbus* dan *Function Codes* [9]

Tipe Data	<i>Absolute Address</i>	<i>Relative Address</i>	<i>Function Code</i>	Deskripsi
<i>Coils</i>	00001 to 09999	0 to 9998	01	Membaca status <i>coil</i>
<i>Discrete Inputs</i>	10001 to 19999	0 to 9998	02	Membaca status <i>input</i>
<i>Input Registers</i>	30001 to 39999	0 to 9998	04	Membaca <i>input register</i>
<i>Holding Register</i>	40001 to 49999	0 to 9998	06	Menulis <i>single register</i>

c. *Data Field Message Framing*

Data field pada *query* berisi kode sebagai informasi tambahan pada *function code* tentang aksi yang harus dikerjakan *slave*.

Informasi tersebut bisa berupa alamat *input-output*, jumlah *input-output*, jumlah byte data, atau nilai data pengesetan. Jika tidak terjadi kesalahan, *data field* pada respon berisi data yang diminta. Sedangkan pada *exception response*, *data field* berisi *exception code*.

d. *Error Checking Field*

Pada mode ASCII, *error checking field* berisi 2 karakter ASCII yang didasarkan pada metode LRC. Prosedur perhitungan nilai LRC adalah

- i. Tambahkan semua *byte* pesan tanpa mengikutkan karakter *start* yaitu *colon* dan karakter *end* yaitu CRLF, dan tanpa melibatkan *carry*.
- ii. Kurangkan nilai FF hex dengan nilai hasil penjumlahan semua *byte* pesan, untuk menghasilkan komplemen 1.
- iii. Tambahkan hasilnya dengan 1 untuk menghasilkan komplemen dua. Hasilnya merupakan nilai LRC.

Pada mode RTU, *error checking field* berisi sebuah nilai 16 bit (2 *byte*) yang didasarkan pada metode CRC. Prosedur perhitungan CRC adalah.

- i. Inisialisasi nilai *register* 16 bit CRC dengan FFFF hex.
- ii. Eksklusif OR 8 bit data pesan pertama dengan *low order byte register* CRC, letakkan hasilnya di-*register* CRC.
- iii. Geser kanan *register* CRC 1 bit ke arah LSB, dan MSB diisi dengan 0. Nilai LSB *register* CRC yang tergeser diperiksa.
- iv. Jika LSB tergeser adalah 0, ulangi langkah 3 (pergeseran yang lain). Jika LSB tergeser 1, eksklusif-OR *register* CRC dengan nilai A001 hex (1010 0000 0000 0001).
- v. Ulangi langkah 3 dan 4 sampai delapan pergeseran. Setelah delapan pergeseran, proses 8 bit data pesan pertama selesai.
- vi. Ulangi langkah 2 - 5 untuk 8 bit data pesan berikutnya sampai semua data diproses.
- vii. Nilai akhir *register* CRC adalah nilai CRC.
- viii. Pada saat CRC dipesan, nilai CRC *low order byte* dikirimkan terlebih dahulu dikuti *high order byte*.

e. *Exception Response*

Terdapat 4 proses komunikasi yang mungkin terjadi antara *master* dan *slave* [9], yaitu

- i. Jika *slave* menerima pesan *query* tanpa adanya kesalahan komunikasi, dan *slave* dapat menangani *query* tersebut, *slave* akan memberikan sebuah respon normal.
- ii. Jika *slave* tidak menerima *query* dikarenakan adanya kesalahan komunikasi, maka tidak ada respon yang dikirimkan. *Master* akan memberikan kondisi *time-out* untuk pengiriman *query* tersebut.
- iii. Jika *slave* menerima pesan *query*, tetapi terdeteksi kesalahan komunikasi (*parity*, LRC, atau CRC), maka tidak ada respon yang dikirimkan. *Master* akan memberikan kondisi *time-out*.
- iv. Jika *slave* menerima *query* tanpa adanya kesalahan komunikasi, tetapi *slave* tidak dapat menangani perintah tersebut (contoh, perintah untuk membaca *coil* atau *register* yang tidak ada), *slave* akan mengirimkan sebuah respon pengecualian (*exception response*) untuk memberikan informasi kepada *master* letak kesalahan yang terjadi.

Sebuah *exception response* mengembalikan kode fungsi dengan MSB (*Most Significant Bit*) yang di-*set* “1” dan *data field* yang diisi dengan kode pengecualian (*exception code*). Hal ini dimaksudkan agar *master* mengetahui *exception* yang terjadi. Beberapa *exception code* berikut keterangannya ditunjukkan pada Tabel 2.3.

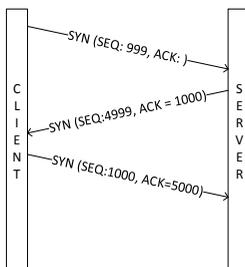
Tabel 2.3 *Exception Code* dalam *Exception Response* [9]

Kode	Nama	Arti
01	<i>Illegal Function</i>	Kode fungsi yang terdapat dalam <i>query</i> merupakan perintah yang tidak diizinkan untuk <i>slave</i>
02	<i>Illegal Data Address</i>	Alamat data dalam <i>query</i> merupakan alamat yang tidak diizinkan untuk <i>slave</i>
03	<i>Illegal Data Value</i>	Nilai dalam <i>data field query</i> merupakan nilai yang tidak diizinkan untuk <i>slave</i>

2.7.2 Modbus TCP/IP

Protokol jenis ini berbasis *connection-oriented protocol* dimana menggunakan satu alamat yang spesifik untuk menghubungkan antar komputer (*source* dan *destination*). Protokol ini mengutamakan kehandalan dalam pengiriman pesan atau informasi yang dikirimkan. Kehandalannya diukur dengan memastikan semua data atau informasi yang dikirimkan dapat tersampaikan dalam bentuk laporan atau tanda terima yang dinamakan *acknowledgment* atau ACK.

Penggunaan TCP memiliki bentuk komunikasi *server – client*. Peran *server* menyediakan layanan data yang dibutuhkan operator sedangkan *client* bertindak sebagai *user* untuk meminta data dari *server*. Ilustrasi proses serah terima data yang terjadi antara *client* dan *server* diawali dengan *client* mengirimkan permintaan data ke *server* seperti pada Gambar 2.14. Permintaan data oleh *client* diterima *server* kemudian *server* merespon permintaan *client* dengan mengirimkan tanda terima (ACK) sebagai laporan telah diterimanya permintaan dan *server* juga mengirimkan informasi/data yang diminta oleh *client*.



Gambar 2.14 Diagram Alur Komunikasi TCP [9]

Keterangan :

SYN : urutan nomer

ACK : *acknowledgment*, laporan aatau tanda serah terima

Modbus TCP/IP merupakan gabungan Modbus dengan RTU dengan antarmuka TCP yang berjalan pada Ethernet. TCP/IP mengacu pada *Transmission Control Protocol* dan *Internet Protocol* yang menyediakan media transmisi untuk pesan Modbus TCP/ IP.

Fungsi utama dari TCP adalah untuk memastikan bahwa semua paket data yang diterima dengan benar, sedangkan IP memastikan bahwa

pesan ditangani dan disalurkan dengan benar. Jadi, Modbus TCP/IP menggunakan TCP/IP dan Ethernet untuk membawa data dari struktur pesan Modbus antara perangkat yang kompatibel. Pada Gambar 2.15, pesan Modbus TCP/IP adalah komunikasi Modbus dikemas dalam sebuah Ethernet TCP/IP.

Modbus Application Protocol (MBAP) Header				Protocol Data Unit (PDU)	
Transaction Identifier	Protocol Identifier	Length Field	Unit ID	Function Code	Data
2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	Varies

Gambar 2.15 Paket Data Modbus TCP [9]

Penjelasan pada Gambar 2.15 diuraikan sebagai berikut.

- a. *Transaction/invocation Identifier (2 bytes):*
Merupakan bidang identifikasi yang digunakan untuk pasangan saat melakukan komunikasi. Ketika beberapa pesan yang dikirim sepanjang koneksi TCP yang sama dengan *client* tanpa menunggu respon.
- b. *Protocol Identifier (2 bytes):*
Bidang ini selalu dalam kondisi “0” untuk layanan Modbus dan nilai-nilai lainnya.
- c. *Length (2 bytes) :*
Frame ini merupakan hitungan *byte* dari bidang yang tersisa dan termasuk *unit identifier byte*, kode fungsi *byte*, dan bidang data.
- d. *Unit Identifier (1 byte):*
Bidang ini digunakan untuk mengidentifikasi *server* yang terletak pada jaringan TCP/IP. Dalam aplikasi *server* Modbus TCP/IP, unit ID diatur ke 00 atau FF atau diabaikan oleh *server*.

Modbus TCP/IP *Application Data Unit* termasuk kedalam frame data TCP dan dikirim melalui ke *port* 502 dimana secara khusus disediakan untuk aplikasi Modbus. *Client* dan *server* pada Modbus TCP/IP mendengarkan dan menerima data melalui *port* 502.

2.7.3 Ethernet

Sebuah jaringan menggunakan metode transmisi *Baseband* yang mengirim sinyalnya secara serial tiap 1 bit pada satu waktu[10]. Teknologi Ethernet digunakan untuk jaringan-jaringan komputer dalam wilayah *Local*. Sistem komunikasi melalui Ethernet membagi aliran data menjadi potongan-potongan pendek yang disebut sebagai bingkai atau *frame*. Setiap bingkai berisi alamat sumber dan tujuan, data dan *error-checking* sehingga data yang rusak dapat dilacak dan dikirimkan kembali. Berdasarkan aturan pada OSI *layer*, Ethernet menyediakan layanan sampai *layer data link*.

a. Jenis-Jenis Ethernet

Berdasarkan dari tingkat kemampuan kecepatan pengiriman data Ethernet terbagi menjadi empat jenis, antara lain sebagai berikut.

- i. 10 Mbit/detik, yang sering disebut sebagai Ethernet. Contoh standar yang digunakan seperti *10Base2*, *10Base5*, *10BaseT*, *10BaseF*.
- ii. 100 Mbit/detik, yang sering disebut sebagai *Fast* Ethernet. Contohnya standarnya seperti *100BaseFX*, *100BaseT*, *100BaseT4*, *100BaseTX*.
- iii. 1000 Mbit/detik atau 1 Gbit/detik, yang sering disebut sebagai *Gigabit* Ethernet.
(standar yang digunakan: *1000BaseCX*, *1000BaseLX*, *1000BaseSX*, *1000BaseT*).

b. Cara Kerja Ethernet

Ethernet beroperasi dalam mode *half-duplex* yang artinya setiap *station* dapat menerima atau mengirim data tapi tidak dapat melakukan keduanya secara sekaligus. *Fast* Ethernet serta Gigabit Ethernet dapat bekerja dalam mode *full-duplex* maupun *half-duplex* seperti pada Gambar 2.16.



Gambar 2.16 Mode Komunikasi *Full Duplex* [10]

Setiap komputer akan "mendengar" terlebih dahulu sebelum "berbicara", artinya mereka akan melihat kondisi jaringan apakah tidak ada komputer lain yang sedang mentransmisikan data. Jika tidak ada komputer yang sedang mentransmisikan data, maka setiap komputer yang ingin mengirimkan data dapat mencoba untuk mengambil alih jaringan dalam mentransmisikan sinyal sehingga dapat dikatakan bahwa jaringan yang menggunakan teknologi Ethernet adalah jaringan yang dibuat berdasarkan basis *First-In-First-Out*.

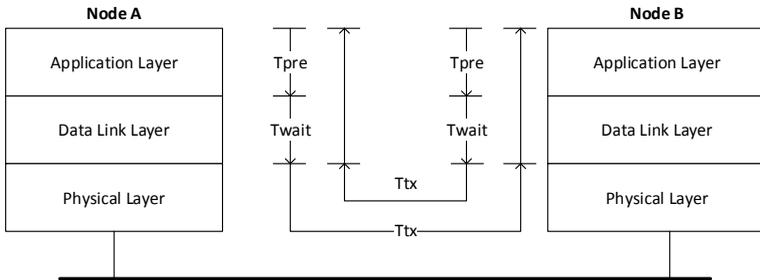
Jika dua *station* hendak mencoba untuk mentransmisikan data pada waktu yang sama, maka kemungkinan akan terjadi *collision* yang akan mengakibatkan dua *station* tersebut menghentikan transmisi data, sebelum akhirnya mencoba untuk mengirimkannya lagi pada interval waktu yang acak (dalam satuan milidetik). Semakin banyak *station* dalam sebuah jaringan Ethernet mengakibatkan jumlah *collision* yang terjadi semakin besar pula dan kinerja jaringan pun akan menjadi buruk.

2.8 Performansi Jaringan

Penggunaan protokol transmisi data yang sesuai diperlukan untuk menjamin kualitas jaringan[11]. Kualitas layanan merupakan kemampuan jaringan agar dapat beroperasi sesuai yang diharapkan. Kontrol jaringan menyediakan batasan terhadap waktu tunda dan jaminan transmisi dari setiap pengiriman data. Parameter-parameter yang dapat digunakan untuk mengetahui performa kontrol jaringan antara lain.

2.8.1 Waktu Tunda

Pada Gambar 2.17, pengiriman paket data dalam jaringan membutuhkan waktu untuk melakukan perjalanan dari satu komputer ke komputer lain (*source* → *destination*). Lamanya waktu tunda dipengaruhi oleh waktu tunda akibat peralatan dan waktu transmisi data yang melintasi jaringan[2]. Pengukuran waktu tunda dilakukan dengan menghitung waktu yang dibutuhkan dari komputer (sumber) ke komputer (tujuan) dalam satu periode.



Gambar 2.17 Diagram Waktu Pengiriman Data [11]

$$\begin{aligned}
 T_{delay} &= T_{destination} + T_{source} \\
 &= T_{preprocessing} + T_{wait} + T_{transmit} + T_{postprocessing}
 \end{aligned}$$

Adapun untuk menghitung total waktu tunda dapat dilakukan menggunakan Persamaan 2.2 yaitu menjumlahkan waktu tunda yang dibutuhkan masing-masing kanal dalam suatu jaringan.

$$T_{delay}^{sum} = \sum_{i \in n_{node}} \sum_{j=1}^{M^{(i)}} T_{delay}^{(i,j)} \quad 2.2$$

Keterangan,

N_{node} merupakan total *node* dalam jaringan.

$M^{(i)}$ merupakan paket data yang diminta oleh kanal sumber

$T_{delay}^{(i,j)}$ merupakan waktu tunda dari kanal sumber ke kanal tujuan.

2.8.2 *Bandwidth*

Merupakan suatu ukuran rentang frekuensi maksimum yang dapat mengalir data dari suatu tempat ke tempat lain dalam suatu waktu tertentu. *Bandwidth* dapat digunakan untuk mengukur aliran data analog maupun aliran data digital. Satuan yang dipakai untuk *bandwidth* adalah *bits per second* atau sering disingkat sebagai bps. *Bit* atau *binary digit* adalah basis angka yang terdiri dari angka 0 dan 1. Satuan ini menggambarkan seberapa banyak bit (angka 0 dan 1) yang dapat mengalir dari satu tempat ke tempat yang lain dalam setiap detiknya melalui suatu media.

Bandwidth adalah konsep pengukuran penting dalam jaringan, tetapi memiliki kekurangan yaitu tidak memperdulikan cara mengirimkan

informasi maupun media apa yang dipakai dalam mengirim informasi. Ini menyebabkan batasan terhadap panjang media yang dipakai, kecepatan maksimal yang dapat dipakai, maupun perlakuan khusus terhadap media yang dipakai [10].

Bandwidth internet disediakan oleh provider internet dengan jumlah tertentu tergantung sewa pelanggan. Adanya QoS dapat diatur agar user tidak menghabiskan *bandwidth* yang di sediakan oleh *provider*. *Bandwidth* mempresentasikan jarak keseluruhan atau jangkauan di antara sinyal tertinggi dan terendah pada kanal (*band*) komunikasi. Pada dasarnya *bandwidth* mempresentasikan kapasitas dari koneksi, semakin tinggi kapasitas, umumnya akan diikuti oleh kinerja yang lebih baik, meskipun kinerja keseluruhan juga tergantung pada faktor-faktor lain, misalnya *latency* yaitu waktu tunda antara masa sebuah perangkat meminta akses ke jaringan dan masa perangkat itu memberi akses untuk melakukan transmisi.

2.8.3 Network Throughput

Merupakan representasi dari laju data yang tersedia dalam jaringan pada saat tersebut. Melalui parameter ini dapat mengetahui tentang kondisi jaringan yang berdasarkan dari perbandingan antara total waktu pengiriman data (dari *node* asal ke *node* tujuan) untuk setiap *node* yang ada dengan total waktu tunda seperti pada Persamaan 2.3.

$$P_{\text{eff}} = \frac{\sum_{i \in N_{\text{node}}} \sum_{j=1}^{M^{(i)}} T_{\text{tx}}^{(i,j)}}{T_{\text{delay}}^{\text{sum}}} \quad 2.3$$

Keterangan,

N_{node} merupakan sekumpulan *node* dalam jaringan.

$M^{(i)}$ merupakan jumlah paket data yang diminta oleh *node* ke-i.

$T_{\text{tx}}^{(i,j)}$ merupakan waktu yang dibutuhkan untuk mengirimkan data dari *node* sumber ke *node* tujuan.

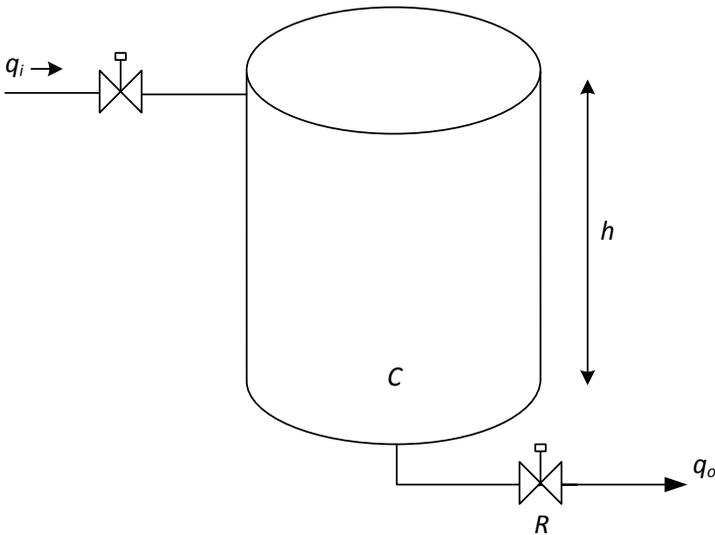
$T_{\text{delay}}^{\text{sum}}$ merupakan total waktu tunda yang terjadi dalam jaringan.

$P_{\text{eff}} \rightarrow 1$ menyatakan bahwa waktu tunda disebabkan sepenuhnya oleh waktu saat proses transmisi data dan dapat dikatakan bahwa performa jaringan dalam keadaan baik $P_{\text{eff}} \rightarrow 0$ dapat dikatakan bahwa terjadi mengalami tumbukan data.

BAB III PERANCANGAN SISTEM

3.1 Pemodelan Sistem Pengaturan *Level*

Sistem pengaturan untuk *level* pada proses penimbunan CPO melibatkan beberapa komponen diantaranya pipa, tangki, sensor dan aktuator. Sistem pengaturan diperlukan untuk menjaga ketinggian *level* CPO agar menjaga temperatur CPO tetap pada kondisi antara 50 sampai 55°C. Secara umum, tangki timbun dan parameternya diilustrasikan pada Gambar 3.1.

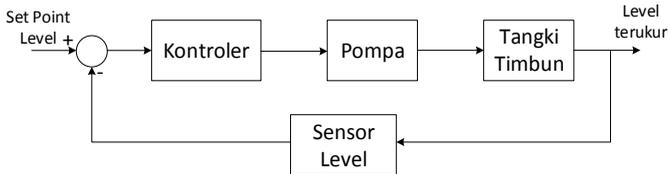


Gambar 3.1 Parameter-Parameter *Level* Tangki CPO

Aktuator pada tangki berupa pompa yang digerakkan dengan masukan tegangan sebesar 0 - 10 V. Adanya tegangan membuat pompa membuka dari 0 - 100% untuk mengalirkan CPO melalui pipa menuju tangki timbun.

Pada tangki terdapat sensor *level* radar yang dapat mendeteksi *level* CPO. Sensor membaca *level* pada tangki apakah sudah sesuai dengan *set point* yang berupa *level* yang diinginkan sesuai berdasarkan Gambar 3.2. Jika nilai yang terbaca oleh sensor kurang dari *set point* maka

kontroler mengatur bukaan pompa agar mengalirkan CPO sesuai dengan *set point* dan waktu yang telah ditentukan.



Gambar 3.2 Diagram Blok Sistem Pengaturan *Level* pada Tangki CPO [4]

Berdasarkan hasil identifikasi, diperoleh parameter-parameter sistem pada Tabel 3.1. Parameter-parameter tersebut digunakan untuk perhitungan pemodelan sistem.

Tabel 3.1 Parameter Sistem Pengaturan *Level* pada Tangki CPO [4]

No.	Parameter	Nilai
1	$q_i = q_o$ (debit masukan dan keluaran)	50 m ³ /jam
2	h (<i>level</i> tangki pada keadaan tunak)	14,66 m
3	D (diameter tangki)	21,35 m
4	C (luas penampang tangki)	357,82 m ²
5	R (perubahan <i>level</i> tiap perubahan debit keluaran)	0,29 jam/m ²
6	η (luas efektif penerima antena sensor) = C	357,82 m ²
7	λ (panjang gelombang yang ditransmisikan sensor) = h	14,66 m
8	d (diameter antena sensor)	0,8 m

3.1.1 Pemodelan *Gain* Pompa [4]

Pada sistem terdapat aktuator berupa pompa di mana masukannya berupa tegangan dan keluarannya berupa debit. Agar masukan pompa berupa satuan debit, maka dilakukan konversi dari tegangan ke debit. Pada model, konversi dari tegangan ke debit dimasukkan sebagai *gain* pompa seperti pada Persamaan 3.1.

$$K_{\text{pompa}} = \frac{50 \text{ m}^3/\text{jam}}{10-0 \text{ V}} = 5 \frac{\text{m}^3/\text{jam}}{\text{V}} \quad 3.1$$

3.1.2 Pemodelan Waktu Tunda karena Pipa [4]

Terdapat pipa juga pada sistem untuk mengalirkan CPO menuju tangki timbun. Karena CPO sebanyak 5.000 ton disuplai dalam waktu 5 *shift* yaitu 40 jam (1 *shift* = 8 jam), maka harus disediakan kapasitas *loading* 50 m³/jam. Ukuran pipa yang digunakan untuk suplai ke dalam tangki timbun adalah 6 *inch* dan jarak dari pabrik kalapa sawit ke tangki timbun CPO adalah 350 m. Pemodelan untuk pipa ini biasa disebut pemodelan waktu tunda yang dapat dimodelkan seperti pada Persamaan 3.2.

$$G_{pipa}(s) = e^{-0,14s} \cong \frac{1}{0,14s+1} \quad 3.2$$

3.1.3 Pemodelan Tangki CPO [4]

Prinsip kerja dari pengaturan *level* adalah menjaga agar debit yang masuk ke tangki (q_i) dan debit yang keluar dari tangki (q_o) besarnya sama (sesuai dengan hukum kekekalan massa), sehingga hubungannya terlihat seperti pada Persamaan 3.3. Parameter-parameter yang terdapat pada tangki CPO dapat dilihat pada Gambar 3.1.

$$q = q_i - q_o \quad 3.3$$

Besar dari q merupakan debit saat kondisi *level* dalam keadaan tunak. Karena debit cairan tiap perubahan waktu sama dengan volume cairan tiap perubahan *level*, Persamaan 3.3 dapat ditulis kembali menjadi Persamaan 3.4.

$$C dh = (q_i - q_o) dt \quad 3.4$$

Nilai C sama dengan nilai luas penampang tangki. Jika luas penampang tangki konstan, maka C juga konstan. Pada sistem terdapat pompa keluaran dengan nilai R yang merupakan perubahan *level* tiap perubahan debit keluaran. Hubungan antara q_o dengan h sesuai definisi dari R dapat dituliskan dalam Persamaan 3.5.

$$q_o = \frac{h}{R} \quad 3.5$$

Di mana h merupakan *level* cairan saat keadaan tunak, jadi nilai $q_i = q_o$. Dengan mensubstitusikan Persamaan 3.5 ke dalam Persamaan 3.4, diperoleh Persamaan 3.6.

$$C \, dh = (q_i - \frac{h}{R}) \, dt \quad 3.6$$

Dengan mengalikan R pada kedua sisi, diperoleh Persamaan 3.7.

$$RC \, dh = (Rq_i - h) \, dt$$

$$RC \frac{dh}{dt} = (Rq_i - h)$$

$$RC \frac{dh}{dt} + h = Rq_i \quad 3.7$$

RC merupakan waktu konstan sistem, sehingga Persamaan 3.7 dilakukan Transformasi Laplace, sehingga menjadi Persamaan 3.8.

$$(RCs+1)H(s) = RQ_i(s) \quad 3.8$$

Jika q_i merupakan masukan dan h keluaran, fungsi alih sistem menjadi Persamaan 3.9.

$$\frac{H(s)}{Q_i(s)} = \frac{R}{RCs+1} \quad 3.9$$

Dengan memasukkan nilai R dan C , diperoleh fungsi alih sistem tangki timbun seperti Persamaan 3.10.

$$\frac{H(s)}{Q_i(s)} = \frac{0,29}{103,77s+1} \quad 3.10$$

3.1.4 Pemodelan Gain Sensor [4]

Pada sistem terdapat sensor radar yang digunakan untuk mendeteksi *level* CPO pada tangki timbun. Parameter untuk menghitung gain antenna diperoleh dari Tabel 3.1.

$$K_{sensor} = \frac{10 - 0}{\eta \left(\frac{\pi \times d}{\lambda} \right)^2} = \frac{10}{357,82 \left(\frac{3,14 \times 0,8}{14,66} \right)^2} = \frac{10}{10,5} = 0,95 \text{ V / m}$$

Dengan menggabungkan seluruh elemen *open-loop* pada sistem, fungsi alih dapat dihitung melalui Persamaan 3.11.

$$G(s) = K_{pompa} \times G_{pipa}(s) \times \frac{H(s)}{Q_i(s)} \quad 3.11$$

Dengan demikian, diperoleh fungsi alih dari *open-loop* sistem seperti pada Persamaan 3.12.

$$G(s) = \frac{1,45}{14,53s^2 + 103,91s + 1} \quad 3.12$$

Dikarenakan respon sistem meyerupai respon orde pertama, maka fungsi alih yang dinyatakan dalam Persamaan 3.13 dapat direduksi menjadi orde pertama, menjadi:

$$G(s) = \frac{1,45}{104s + 1} \quad 3.13$$

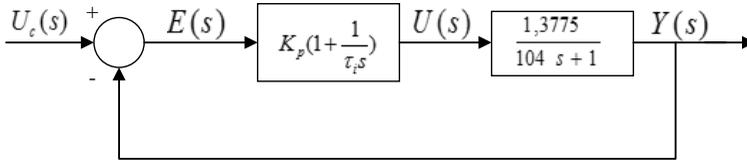
Berdasarkan karakteristik yang diperoleh dari respon sistem *open-loop* yaitu $E_{ss} = 45\%$ dan *time constant* yang cukup lama yaitu 104 jam, diperlukan merancang kontroler dengan spesifikasi untuk meminimalkan E_{ss} dan mempercepat *time constant*.

3.1.5 Perancangan Kontroler PI

Metode analitik ini digunakan untuk mendesain kontroler PI dengan menentukan spesifikasi respon yang diinginkan yaitu $E_{ss} = 0\%$ dan respon 5x lebih cepat. Dirancang sistem *closed-loop* dengan cara mengalikan *gain* sensor sebesar 0,95 dengan fungsi alih sistem *open-loop* pada Persamaan 3.14, sehingga fungsi alihnya menjadi:

$$G(s) = \frac{1,3775}{104s + 1} \quad 3.14$$

Diagram blok sistem *closed-loop* pengaturan *level* pada tangki CPO ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram Blok Sistem dengan Kontroler PI

Dari hasil perhitungan fungsi alih tersebut, diperoleh parameter sistem orde pertama dengan $K = 1.3775$ dan $\tau = 104 \text{ jam}$. Apabila diinginkan spesifikasi sistem dengan parameter $K^* = 1$ dan $\tau^* = 21 \text{ jam}$, maka perlu dirancang kontroler PI seperti yang dijelaskan pada Gambar 3.3. Dapat dilihat hubungan antara masukan $U_c(s)$ dan keluaran $Y(s)$ untuk mendapatkan parameter PI.

$$\frac{Y(s)}{U_c(s)} = \frac{K_P \frac{\tau_i s + 1}{\tau_i s} \times \frac{1,3775}{104s + 1}}{1 + (K_P \frac{\tau_i s + 1}{\tau_i s} \times \frac{1,3775}{104s + 1})}$$

Dengan membandingkan hubungan antara $U_c(s)$ dan $Y(s)$ dengan fungsi alih spesifikasi desain sebagai berikut.

$$\frac{Y^*(s)}{U_c^*(s)} = \frac{1}{21s + 1}$$

Maka diperoleh τ_i sebesar

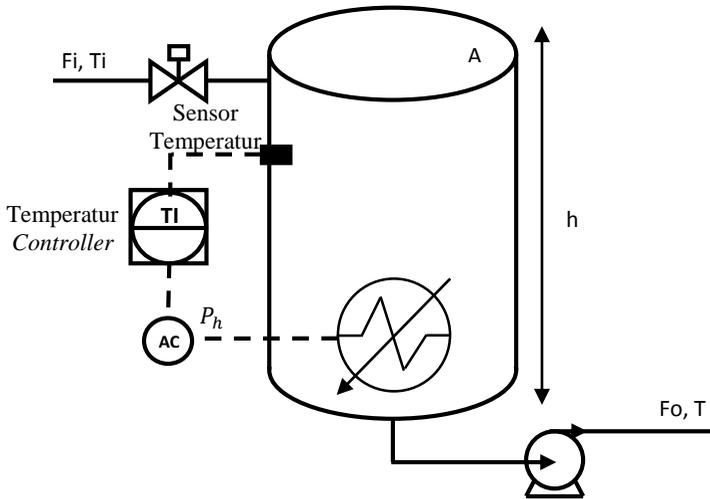
$$\tau_i = \tau = 104$$

Setelah memasukkan nilai τ_i , diperoleh nilai K_p :

$$K_p = \frac{4,95}{1,3775} = 3,6$$

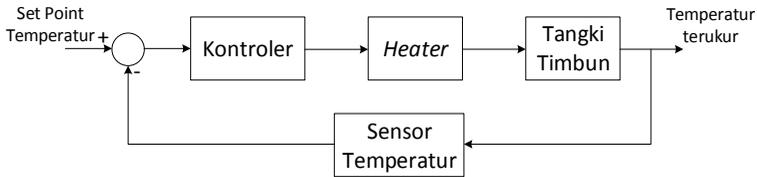
3.2 Pemodelan Sistem Pengaturan Temperatur [4]

Pada tangki timbun diperlukan sistem pengaturan temperatur untuk menjaga agar CPO tetap pada temperatur 50 – 55°C. Tangki juga dilengkapi dengan pipa pemanas dan sensor untuk mengukur temperatur CPO. Lebih lengkap proses pemodelan yang terlibat mengenai sistem Tangki CPO dijelaskan seperti pada Gambar 3.4.



Gambar 3.4 Parameter-Parameter Temperatur Tangki CPO [4]

Sistem pengaturan temperatur melibatkan komponen sensor, kontroler, dan aktuator seperti diagram blok pada Gambar 3.5. Pembacaan sensor untuk mengetahui besar temperatur CPO pada tangki. Apabila nilai yang terbaca tidak sesuai dengan *set point* yang dimasukkan, kontroler akan mengirimkan sinyal perbaikan ke *plant*. Pemanas akan mengatur set keluaran panas agar sesuai dengan *set point*.



Gambar 3.5 Diagram Blok Sistem Pengaturan Temperatur pada Tangki CPO [4]

Aliran masuk CPO dengan temperatur yang bervariasi akan ditampung dalam tangki timbun. Penambahan temperatur CPO dalam tangki timbun dilakukan menggunakan pemanas elektrik. Ketika temperatur terukur masih kurang dari 50°C, maka kontroler akan mengatur daya elemen pemanas untuk memanaskan CPO sesuai kebutuhan. Parameter-parameter yang digunakan untuk sistem pengaturan temperatur yang dirancang tercantum pada Tabel 3.2.

Tabel 3.2 Parameter-Parameter Sistem Pengaturan Temperatur Tangki CPO [4]

Simbol	Keterangan	Satuan
F_i	Aliran cairan yang masuk ke tangki	m^3/jam
F_o	Aliran cairan yang keluar dari tangki	m^3/jam
T_i	Temperatur <i>input</i>	$^{\circ}C$
T	Temperatur <i>output</i>	$^{\circ}C$
A	Luas alas tangki	m^2
h	Tinggi <i>level</i> cairan (dengan asumsi cairan dalam tangki penuh dan konstan)	m
V	Volume tangki	m^3
m	Massa cairan	Kg
ρ	Massa jenis zat cair	Kg/m^3
W_{in}	Energi masuk	J
W_{out}	Energi keluar	J
C_v	Kalor jenis zat cair dengan volume konstan	$J/kg\ ^{\circ}C$

Pada sistem ini model matematika *plant* diperoleh berdasarkan hukum-hukum fisisnya. Hasil pemodelan *plant* seperti pada Persamaan 3.25. Berdasarkan hukum termodinamika, kalor dalam tangki dapat dirumuskan dengan Persamaan 3.15.

$$W = m.Cv.\Delta T = m.Cv.(T - Ti) \quad 3.15$$

Berdasarkan hukum kesetimbangan energi, berlaku bahwa akumulasi energi dalam tangki sama dengan energi yang masuk dikurangi energi keluar ditambahkan dengan usaha dari luar.

$$m.Cv.\frac{dT}{dt} = Win - Wout + K_h P_h \quad 3.16$$

Pada saat terjadi aliran cairan masuk dan keluar tangki, volume diabaikan sehingga didapatkan Persamaan 3.17.

$$\frac{A.h}{Fo} \cdot \frac{dT}{dt} = \frac{Fi}{Fo} Ti - T + \frac{K_h P_h}{\rho.Cv Fo} \quad 3.17$$

Pengaturan temperatur terjadi saat *level* cairan di dalam tangki penuh dan dijaga tetap. Berdasarkan kondisi tersebut diperoleh maka diperoleh Persamaan 3.18.

$$\frac{A.h}{Fo} \cdot \frac{dT}{dt} = Ti - T + \frac{K_h P_h}{\rho.Cv Fo} \quad 3.18$$

Diasumsikan bahwa temperatur dari cairan yang masuk ke dalam tangki (T_i) konstan, sehingga didapatkan Persamaan 3.19.

$$\frac{A.h}{Fo} \cdot \frac{dT}{dt} = -T + \frac{K_h P_h}{\rho.Cv Fo} \quad 3.19$$

Transformasi Laplace dari Persamaan 3.19 adalah sebagai berikut.

$$\frac{A.h}{Fo} \cdot sT(s) + T(s) = \frac{K_h P_h}{\rho.Cv Fo} \cdot P_h(s) \quad 3.20$$

$$\frac{T(s)}{P_h(s)} = \frac{K_h}{\rho \cdot C_v \cdot F_o \left(\frac{V}{F_o}\right)s + 1} \quad 3.21$$

Beberapa parameter diketahui untuk mendapatkan model matematika *plant* berdasarkan hukum fisis. Parameter-parameter yang digunakan dan diketahui nilainya tercantum pada Tabel 3.3.

Tabel 3.3 Parameter Sistem yang Diketahui Nilainya [4]

Simbol	Keterangan	Nilai	Satuan
Fo	Aliran air keluar	50	m ³ /jam
Cv	Kapasitas panas air dengan volume konstan	4.18	J/g °C
ρ	Massa jenis zat cair	1	g/m ³
V	Volume tangki	5251.137	m ³
P _h	<i>Maximum power heating element</i>	840	W
K _h	<i>Heating element power gain</i>	504	J

Dari Tabel 3.3 dapat diperoleh fungsi alih tangki timbun seperti pada Persamaan 3.22.

$$\frac{T(s)}{P_h(s)} = \frac{2.411}{105.022s + 1} \quad 3.22$$

3.2.1 Pemodelan Sensor RTD [4]

Tipe sensor temperatur yang akan digunakan pada sistem ini adalah RTD dengan rentang masukan temperatur 0° sampai 55° C. Keluaran yang dihasilkan berupa resistansi yang akan dikonversi menjadi tegangan antara 0 – 10 Volt. Pada pemodelan ini diperoleh penguatan RTD yang dilambangkan dengan K_{RTD} dimana nilai penguatan tersebut dirumuskan sebagai perbandingan antara tegangan keluaran dengan temperatur cairan dalam tangki.

$$K_{\text{RTD}} = \frac{\text{Tegangan output (Volt)}}{\text{Temperatur (}^{\circ}\text{C)}} \quad 3.23$$

Dari pemodelan di atas diperoleh *gain* sensor yang merupakan Persamaan masukan dan keluarannya seperti pada Persamaan 3.24.

$$K_{\text{RTD}} = \frac{10 \text{ (Volt)}}{55 \text{ (}^{\circ}\text{C)}} \quad 3.24$$

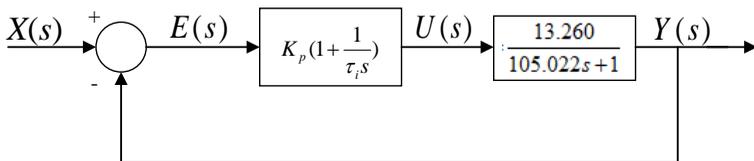
Sistem secara keseluruhan terdiri dari *plant* tangki, pemanas elektriknya dan sensor temperatur sehingga diperoleh sistem secara keseluruhan dengan fungsi alihnya pada Persamaan (3.25).

$$\frac{Y(s)}{X(s)} = \frac{13.260}{105.022s + 1} \quad 3.25$$

3.2.2 Perancangan Kontroler PI

Perancangan kontroler dilakukan berdasarkan kriteria yang ditentukan dan respon sistem *open loop plant* hasil pemodelan berdasarkan hukum fisis untuk merancang kontroler PI yang sesuai dengan spesifikasi yang diinginkan.

Sebelum temperatur diatur, tangki CPO diasumsikan memiliki temperatur ruang sebesar 20°C, sehingga respon sistem tidak muncul dari 0°C tetapi dari nilai *offset* yang diatur yaitu 20°C. Berdasarkan hasil pengujian sistem *open loop* terjadi kesalahan keadaan tunak yang besar antara respon *plant* dengan *set point* yang diberikan. Sistem ini diperbaiki dengan merancang kontroler PI.



Gambar 3.6 Diagram Blok Sistem Pengaturan Temperatur dengan Kontroler PI

Spesifikasi perancangan kontroler PI yaitu tidak ada kesalahan keadaan tunak dan waktu pemanasan untuk mencapai temperatur

maksimal yang awalnya hampir memakan waktu 9 jam yang dalam simulasi ini dipercepat dalam satuan menit menjadi 9 menit. Waktu tersebut dibuat 3 kali lebih cepat menjadi 3 menit. Diagram blok sistem dengan kontroler PI seperti pada Gambar 3.6.

Perancangan kontroler P dengan spesifikasi sistem yang diinginkan adalah respon 3 kali lebih cepat dari respon sistem *open loop* dimana persamaan sistem seperti pada Persamaan 3.26.

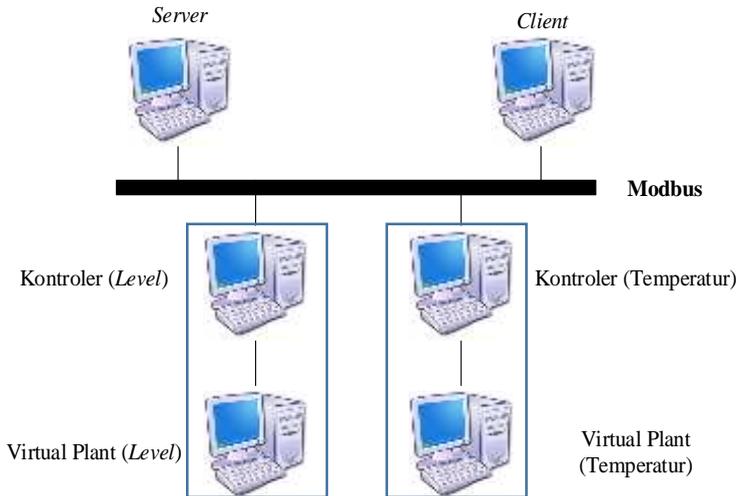
$$\frac{Y(s)}{X(s)} = \frac{13.260}{105.022s + 1} \quad 3.26$$

Berdasarkan spesifikasi respon sistem yang diinginkan, diperoleh $\tau^* = 35.007$, sehingga diperoleh gain proporsional $K_p = 0.15083$. Perancangan kontroler proporsional hanya untuk mempercepat respon sistem, bukan untuk menghilangkan kesalahan keadaan tunak. Oleh karena itu, harus ditambahkan integrator untuk menghilangkan kesalahan keadaan tunak sehingga sistem dapat mencapai set point yang diinginkan. Penambahan integrator dipilih $\tau_i = \tau = 35.007$. Penentuan tersebut bertujuan agar respon hasil perancangan kontroler PI berupa respon orde 1.

3.3 Perancangan Protokol Jaringan Modbus

Pengawasan dan pengendalian pengaturan *level* dan temperatur dilakukan oleh *server* yang terhubung dengan kontroler pada tangki CPO. Komunikasi antara *server* dengan kontroler menggunakan jaringan komunikasi.

Komunikasi berawal dari *plant* yang mengirimkan menuju ke masing-masing kontroler untuk pengaturan *level* dan temperatur. Server akan menampung semua data berupa *set point*, *manipulated value*, *process value* dan lain sebagainya dari masing-masing kontroler. Skema jaringan komunikasi sistem pengaturan *level* dan temperatur pada tangki CPO melibatkan beberapa perangkat seperti pada Gambar 3.7.



Gambar 3.7 Rancangan Arsitektur Jaringan Komunikasi

3.3.1 Perancangan Komunikasi Kontroler dengan *Server* untuk Pengaturan *Level*

Sebuah kontroler telah dirancang yang dapat mengendalikan jalannya proses pengaturan level CPO yang memiliki fungsi alih sistem seperti Persamaan 3.14. Hasil rancangan tersebut didapatkan fungsi alih sistem orde pertama dengan $K = 1.3775$ dan $\tau = 104 \text{ jam}$. Pengiriman data dari kontroler (*level*) ke *server* dilakukan menggunakan protokol Modbus dengan mendeklarasikan variabel-variabel data yang dikirimkan seperti diilustrasikan pada Gambar 3.7.

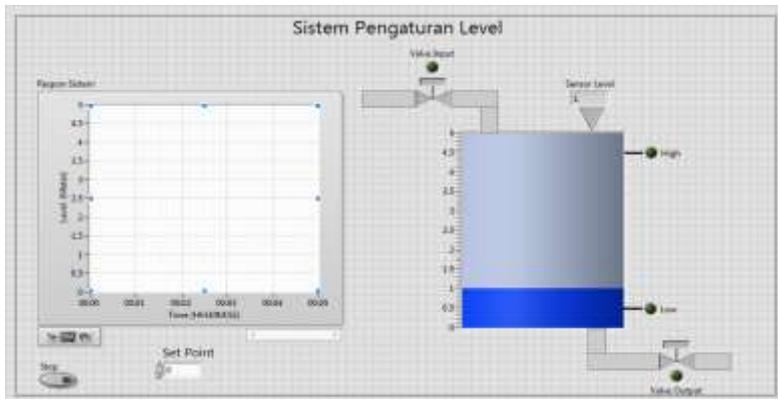
Berdasarkan Tabel 3.4 dijelaskan beberapa *setting* untuk melakukan pengiriman data variabel pengaturan level tangki CPO seperti pembacaan sensor, indikator *level*, dan indikator *valve*.

Tabel 3.4 *Setting* Parameter Modbus untuk Pengaturan *Level*

<i>IP Address</i>		<i>Modbus Address</i>	<i>Function</i>	Data	Variabel
Kontroler	<i>Server</i>				
10.10.10.20	10.10.10.1	00001	<i>Digital input</i>	(1-bit)	Sensor <i>Low Level</i>
		00002		(1-bit)	Sensor <i>High Level</i>

IP Address		Modbus Address	Function	Data	Variabel
Kontroler	Server				
10.10.10.20	10.10.10.1	00003	Digital Output	(1-bit)	Indikator Low Level
		00004		(1-bit)	Indikator High Level
		30001	Analog Input	(16-bit Integer)	Baca Set Point
		30002		(16-bit Integer)	Baca Sinyal Sensor
		30003	Analog Output	(16-bit Integer)	Kirim Sinyal Kontrol

Variabel-variabel pengaturan *level* dideklarasikan dengan melakukan *setting* seperti pada Tabel 3.4 agar dapat diakses oleh *server*. *Server* yang menyajikan data-data tersebut kedalam bentuk HMI agar memudahkan operator dalam melakukan pengawasan dan pengendalian seperti ditampilkan pada Gambar 3.8.



Gambar 3.8 HMI Pengaturan *Level* pada Tangki CPO

Setelah komunikasi antara kontroler (*level*) dengan *server* dirancang, selanjutnya dapat dilakukan pengujian untuk menguji performa jaringan. Pengujian tersebut dilakukan untuk membuktikan bahwa performa jaringan berpengaruh terhadap respon sistem pengaturan *level*.

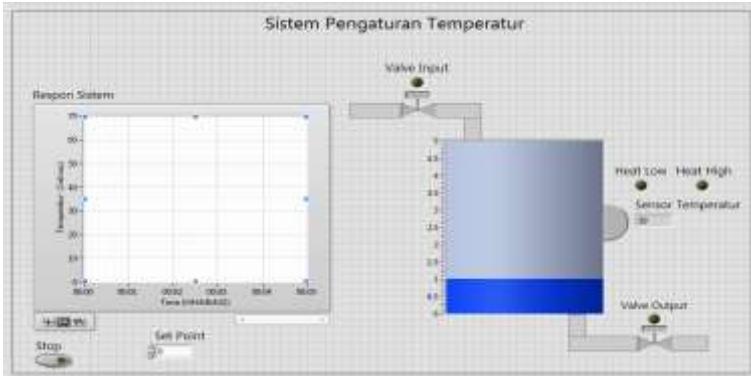
3.3.2 Komunikasi Kontroler dengan Server untuk Pengaturan Temperatur

Sebuah kontroler dirancang untuk mengendalikan jalannya proses pengaturan level CPO yang memiliki fungsi alih sistem seperti Persamaan 3.25. Hasil dari rancangan kontroler diperoleh parameter sistem orde pertama dengan $\tau^*=35.007$ dan *gain* proporsional $K_p=0.15083$. Pengiriman data dari kontroler (temperatur) ke *server* dilakukan menggunakan protokol Modbus dengan mendeklarasikan variabel-variabel data yang dikirimkan seperti diilustrasikan pada Gambar 3.7.

Tabel 3.5 *Setting* Parameter Modbus untuk Pengaturan Temperatur

<i>IP Address</i>		<i>Modbus Address</i>	<i>Function</i>	Data	Variabel
Kontroler	Server				
10.10.10.30	10.10.10.1	00005	<i>Digital input</i>	(1-bit)	Sensor <i>Low Temperatur</i>
		00006		(1-bit)	Sensor <i>High Temperatur</i>
		00007	<i>Digital Output</i>	(1-bit)	Indikator <i>Heat Low</i>
		00008		(1-bit)	Indikator <i>Heat High</i>
		30004	<i>Analog Output</i>	(16-bit <i>Integer</i>)	Kirim Sinyal Kontrol
		30005	<i>Analog Input</i>	(16-bit <i>Integer</i>)	Baca Sinyal Sensor
		30006		(16-bit <i>Integer</i>)	Baca <i>Set Point</i>

Pada Tabel 3.5 menjelaskan beberapa *setting* parameter protokol Modbus untuk melakukan pengiriman data variabel pengaturan temperatur tangki CPO seperti data pembacaan sensor, indikator panas, dan indikator bukaan *valve*. Variabel dideklarasikan dengan melakukan *setting* seperti pada Tabel 3.5 agar dapat diakses oleh *server* maupun *client*. Data kontroler dikirimkan menuju *server* dan disajikan dalam bentuk HMI untuk memudahkan operator dalam melakukan pengawasan dan pengendalian seperti pada Gambar 3.9.

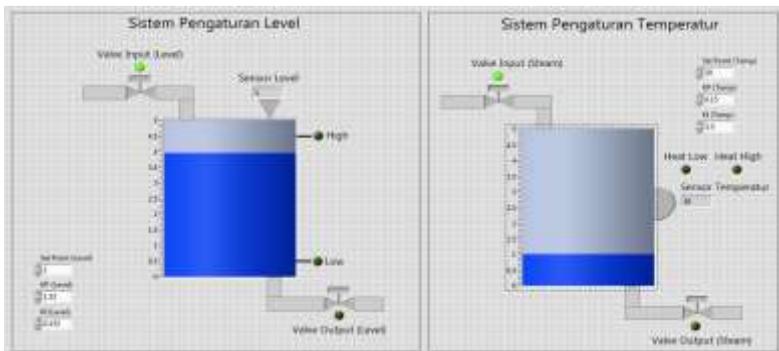


Gambar 3.9 HMI Pengaturan Temperatur pada Tangki CPO

Setelah komunikasi antara kontroler (temperatur) dengan *server* dirancang, selanjutnya dapat dilakukan pengujian untuk menguji performa jaringan. Pengujian tersebut dilakukan untuk membuktikan bahwa performa jaringan berpengaruh terhadap respon sistem pengaturan temperatur.

3.3.3 Perancangan Komunikasi *Server* dengan *Client*

Variabel-variabel yang dikirimkan oleh kontroler *level* dan temperatur dibaca oleh *server* dan diubah kedalam bentuk tampilan HMI seperti pada Gambar 3.10.



Gambar 3.10 HMI *Server* pada Tangki CPO

Pengiriman data dari kontroler (temperatur) ke *server* dilakukan menggunakan protokol Modbus dengan mendeklarasikan variabel-variabel data yang dikirimkan seperti diilustrasikan pada Gambar 3.7. Pada Gambar 3.10 menyajikan data *level* dan temperatur seperti *set point*, *manipulated variable*, *process variable*, indikator *level* dan temperatur dan lain sebagainya. Tahap sebelumnya dilakukan beberapa *setting* parameter Modbus untuk variabel pengaturan *level* dan temperatur pada tangki CPO seperti data pembacaan sensor, *process value*, indikator panas, dan indikator bukaan *valve* sesuai dengan Tabel 3.6.

Tabel 3.6 *Setting* Parameter Modbus untuk Komputer *Server-Client*

<i>IP Address</i>		<i>Modbus Address</i>	<i>Function Code</i>	Data	Variabel
<i>Server</i>	<i>Client</i>				
10.10.10.1	10.10.10.100	30001	<i>Analog Input</i>	(16-bit)	Baca <i>Set Point (Level)</i>
		30002		(16-bit)	Baca Sinyal Sensor (<i>Level</i>)
		30003	<i>Analog Output</i>	(16-bit)	Kirim Sinyal Kontrol (<i>Level</i>)
		30004		(16-bit)	Kirim Sinyal Kontrol (<i>Temp</i>)
		30005	<i>Analog Input</i>	(16-bit)	Baca Sinyal Sensor (<i>Temp</i>)
		30006		(16-bit)	Baca <i>Set Point (Temp)</i>
		00001	<i>Digital Input</i>	(1-bit)	Sensor <i>Low Level</i>
		00002		(1-bit)	Sensor <i>High Level</i>
		00003	<i>Digital Output</i>	(1-bit)	Indikator <i>Low Level</i>
		00004		(1-bit)	Indikator <i>High Level</i>
		00005	<i>Digital Input</i>	(1-bit)	Sensor <i>Low Temperatur</i>

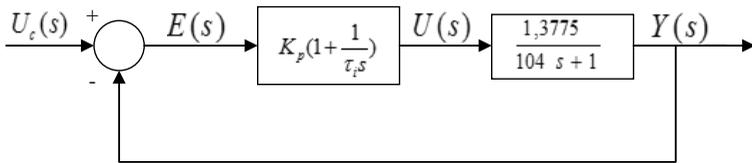
<i>IP Address</i>		<i>Modbus Address</i>	<i>Function Code</i>	<i>Data</i>	<i>Variabel</i>
<i>Server</i>	<i>Client</i>				
10.10.10.1	10.10.10.100	00006	<i>Digital Input</i>	<i>(1-bit)</i>	<i>Sensor High Temperatur</i>
		00007	<i>Digital Output</i>	<i>(1-bit)</i>	<i>Indikator Heat Low</i>
		00008		<i>(1-bit)</i>	<i>Indikator Heat High</i>

Setelah komunikasi antara kontroler (temperatur) dengan *server* dirancang, selanjutnya dapat dilakukan pengujian untuk menguji performa jaringan. Pengujian tersebut dilakukan untuk membuktikan bahwa performa jaringan berpengaruh terhadap respon sistem pengaturan temperatur.

BAB IV PENGUJIAN DAN ANALISA

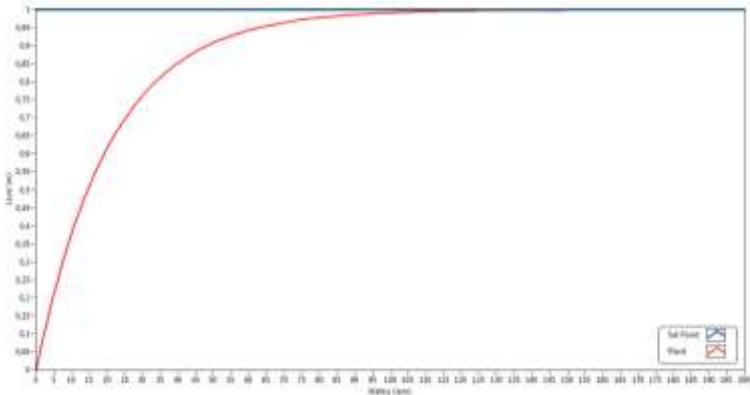
4.1 Pengujian Kontroler PI Sistem Pengaturan *Level*

Berdasarkan hasil perancangan yang dihasilkan maka dilakukan pengujian mengenai kontroler PI untuk sistem pengaturan *level*. Untuk melakukan pengujian tersebut dibangun sistem seperti pada Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem Pengaturan *Level* dengan Kontroler PI

Dengan memasukkan parameter K_p dan K_i ke dalam sistem, diperoleh respon *plant* dengan kontroler PI pada Gambar 4.2.

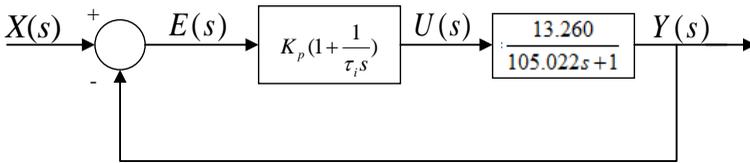


Gambar 4.2 Respon Sistem Pengaturan *Level* dengan Kontroler PI

Berdasarkan respon yang diperoleh pada Gambar 4.2, respon sesuai dengan spesifikasi desain yaitu menjadi 5x lebih cepat dan kesalahan waktu tunak menjadi 0%.

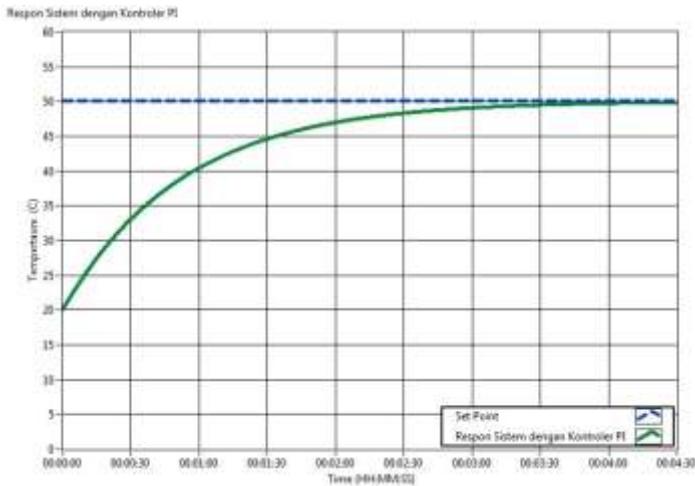
4.2 Pengujian Kontroler PI Sistem Pengaturan Temperatur

Perancangan kontroler PI bertujuan untuk menghilangkan kesalahan keadaan tunak yang terjadi saat sistem hanya diberikan kontroler proporsional. Spesifikasi perancangan kontroler tersebut adalah $\tau_i = \tau = 105.022$ sehingga respon yang dihasilkan berupa respon model orde pertama. *Setting* pengujian sistem dengan kontroler PI adalah seperti pada Gambar 4.3.



Gambar 4.3 Diagram Blok Sistem Pengaturan Temperatur dengan Kontroler PI

Respon sistem dengan penambahan integrator seperti *setting* pengujian Gambar 4.3 dapat dilihat pada Gambar 4.4.



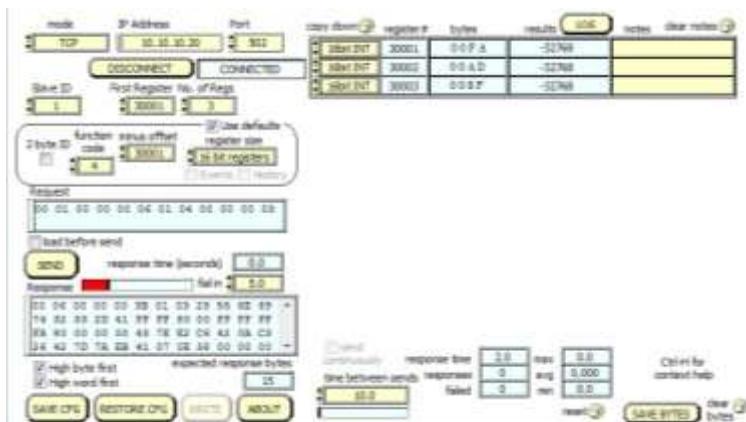
Gambar 4.4 Respon Sistem Pengaturan Temperatur dengan Kontroler PI

Dari Gambar 4.4 diperoleh, dengan penambahan integrator, kesalahan keadaan tunak sistem dapat mencapai 0%. Sistem mencapai

keadaan tunak sesuai *set point* yang diberikan, dalam waktu kurang lebih 3 menit. Waktu yang dibutuhkan sistem tersebut untuk mencapai keadaan tunak sama dengan waktu yang dibutuhkan sistem dengan kontroler proporsional. Perbedaan respon kedua sistem tersebut terletak pada nilai keadaan tunak yang dicapai, sehingga dapat disimpulkan bahwa respon sistem dengan penambahan integrator dapat memenuhi spesifikasi yang ditentukan.

4.3 Pengujian Protokol Jaringan Modbus

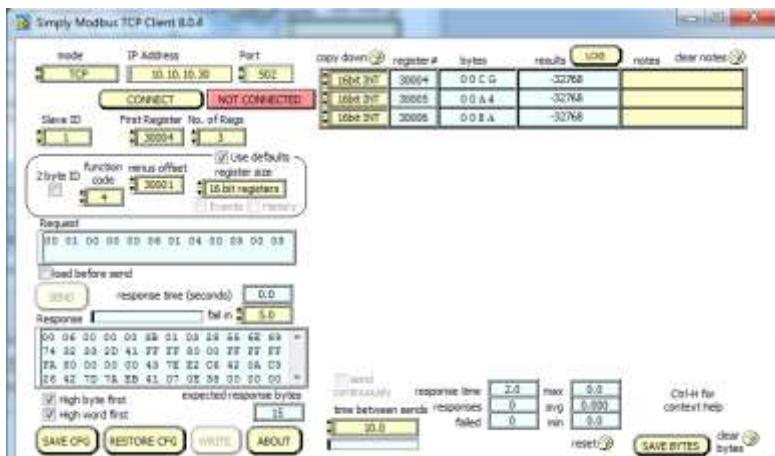
Pada pengujian ini bertujuan untuk mengetahui status komunikasi jaringan Modbus untuk sistem pengaturan *level*. Pengujian dilakukan dengan melakukan konfigurasi seperti yang dijelaskan pada Tabel 3.4, kemudian melihat status komunikasi antara kontroler dengan *server* dan melihat respon dari sistem pengaturan *level*.



Gambar 4.5 Status Komunikasi Modbus pada Sistem Pengaturan *Level*

Berdasarkan Gambar 4.5 bahwa status komunikasi antara kontroler dengan *server* telah terhubung. Masing-masing data yang dikirimkan oleh kontroler dapat terbaca nilainya sesuai dengan alamat *register*. Nilai yang terbaca oleh sensor yang berupa sinyal akan dikonversi kedalam satuan *integer* karena proses pengiriman dilakukan sesuai dengan format pada *register*. Respon waktu yang dibutuhkan *server* dengan kontroler dalam merespon data yang dikirim yaitu sebesar 2 detik.

Pengujian selanjutnya bertujuan untuk mengetahui status komunikasi jaringan Modbus untuk sistem pengaturan temperatur. Pengujian dilakukan dengan melakukan konfigurasi seperti yang dijelaskan pada Tabel 3.5, kemudian melihat status komunikasi antara kontroler dengan *server* dan melihat respon dari sistem pengaturan temperatur.



Gambar 4.6 Status Komunikasi Modbus pada Sistem Pengaturan Temperatur

Pada Gambar 4.6 menunjukkan status komunikasi antara kontroler dengan *server* telah terkoneksi. Data yang dikirimkan oleh kontroler dapat terbaca nilainya sesuai dengan alamat *register*. Nilai yang terbaca oleh sensor yang berupa sinyal akan dikonversi kedalam satuan *integer* karena proses pengiriman dilakukan sesuai dengan format pada *register*. Respon waktu yang dibutuhkan *server* dengan kontroler dalam merespon data yang dikirim yaitu sebesar 2 detik.

Tabel 4.1 Pembacaan Register

Register	Bytes	Data	Keterangan
30001	00FA	16-bit	Baca Set Point (Level)
30002	00AD	16-bit	Baca Sinyal Sensor (Level)
30003	008F	16-bit	Kirim Sinyal Kontrol (Level)
30004	00CG	16-bit	Kirim Sinyal Kontrol (Temp)
30005	00A4	16-bit	Baca Sinyal Sensor (Temp)
30006	008A	16-bit	Baca Set Point (Temp)

Dari kedua pengujian diatas dapat dilihat pada Tabel 4.1 bahwa komunikasi antara *server* dan kontoler berdasarkan alamat IP dan alamat *register*. Setiap pengiriman data yang dilakukan memiliki alamat *register*. Nilai dari setiap data akan dikonversi menjadi satuan *register* yaitu *integer*.

4.4 Pengujian Performa Jaringan Modbus

Berdasarkan hasil perancangan pada Gambar 3.7 dilakukan pengujian terhadap perubahan message size terhadap respon sistem, dan pengaruh jumlah paket data terhadap performa jaringan.

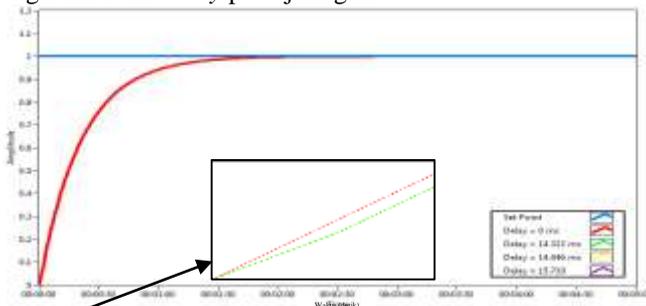
4.4.1 Pengaruh Variasi Message Size

Pengujian ini untuk mengetahui pengaruh ukuran data terhadap performa jaringan dengan respon sistem pengaturan *level* dan temperatur. Perubahan nilai *set point* berpengaruh terhadap besar paket data yang dikirimkan. Pengujian dilakukan dengan memberikan *set point* terhadap sistem pengaturan *level* dan pengaturan temperatur sebesar 1 meter dan 50 derajat celcius. Hasil pengujiannya dapat dilihat pada Tabel 4.2.

Tabel 4.2 Performa Jaringan

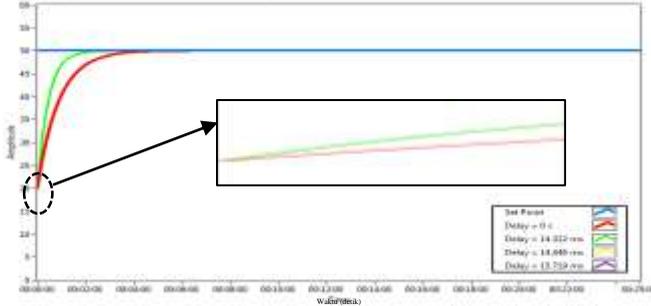
Message Size (bytes)	54	66	71
Waktu tunda (milidetik)	14,322	14,446	15,719
Bandwidth (kbit/detik)	297	297	297
Throughput (%)	55,8	56,1	56,2
Jitter (detik)	0,001	0,001	0,001

Adanya variasi *message size* menyebabkan *delay* mengalami perubahan. Dari Tabel 4.2 terbukti bahwa ukuran data yang dikirimkan mempengaruhi besar *delay* pada jaringan.



Gambar 4.7 Pengaruh Waktu Tunda Jaringan pada Pengaturan Level

Berdasarkan Gambar 4.7, *level* yang diinginkan terisi pada tangki adalah 5 meter. *Delay* awal sistem sebesar 0 detik. Adanya pengaruh *delay* sebesar 40 milidetik pada pengaturan *level* mengakibatkan respon dapat mencapai keadaan tunak pada waktu 2 menit.

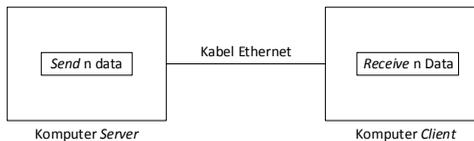


Gambar 4.8 Pengaruh Waktu Tunda pada Pengaturan Temperatur

Dari kedua hasil tersebut dapat dinyatakan bahwa masing-masing memiliki laju pengiriman data yang sama yaitu sebesar 297 kbits/s. Adanya perbedaan *message size* tersebut menyebabkan respon keadaan tunak pada pengaturan *level* dan temperatur mengalami penundaan serta *throughput* atau tingkat keberhasilan pengiriman data dari *server* ke *client* juga mengalami perbedaan.

4.4.2 Pengaruh Variasi Jumlah Paket Data

Pada simulasi ini dilakukan dengan mengirimkan sejumlah paket data dari *server* menuju ke *client*. Pengujian dilakukan untuk mengetahui pengaruh perubahan variasi jumlah paket data yang dikirim terhadap performa jaringan. Perubahan variasi jumlah paket data dilakukan sebanyak 10 kali dengan *range* pengiriman antara 1000 hingga 10000. Pemilihan pada *range* tersebut dilakukan karena untuk pengiriman paket data diasumsikan memiliki banyak jenis data yang dikirim. *Setting* pengujian variasi jumlah paket data seperti pada Gambar 4.9.



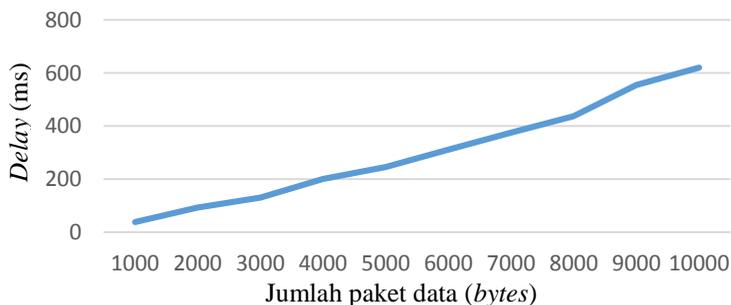
Gambar 4.9 Blok Konfigurasi Pengiriman Paket Data

Dari Tabel 4.3 didapatkan pengaruh perubahan variasi paket terhadap performa jaringan seperti *delay*, kecepatan pengiriman data dan tingkat keberhasilan data yang dikirimkan. Parameter *delay* pada pengujian tersebut mengalami peningkatan karena setiap data yang dikirimkan dari satu komputer menuju komputer lainnya membutuhkan tahapan untuk memproses sinyal menjadi data digital (waktu *pre-processing*), waktu tunggu, waktu untuk melakukan pengiriman data dari satu komputer ke komputer, dan waktu *post-processing* seperti pada Gambar 2.17.

Tabel 4.3 Variasi Jumlah Pengiriman Paket Data

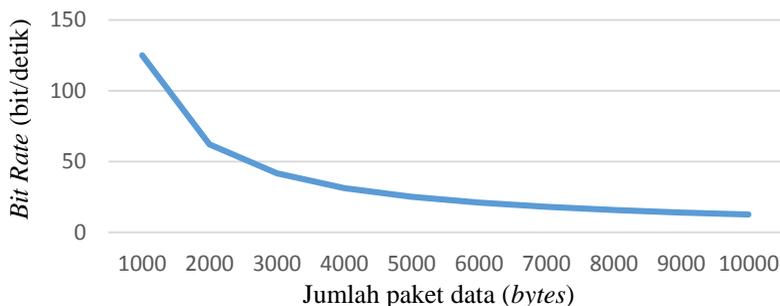
No	Jumlah paket data (bytes)	<i>Delay</i> (ms)	Kecepatan data (bit/detik)	<i>Throughput</i> (%)
1	1000	38	125,00	100
2	2000	93	62,50	100
3	3000	130	41,67	100
4	4000	200	31,25	100
5	5000	245	25,00	100
6	6000	311	20,83	100
7	7000	375	17,86	100
8	8000	437	15,62	100
9	9000	555	13,89	100
10	10000	620	12,50	100

Dari pengujian diatas didapatkan bahwa banyaknya jumlah data yang dikirimkan dalam suatu jaringan menyebabkan waktu yang dibutuhkan untuk melakukan pengiriman data juga semakin lama sehingga menyebabkan munculnya *delay* seperti pada Gambar 4.10.



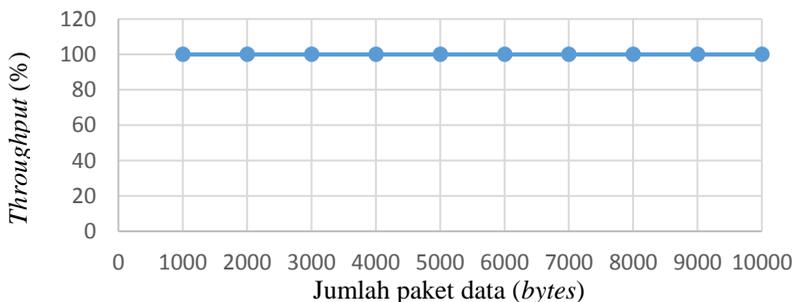
Gambar 4.10 Perbandingan Jumlah Paket data terhadap *Delay*

Pengaruh variasi paket data terhadap performa kecepatan pengiriman data dapat dilihat hasilnya pada Gambar 4.11. Parameter ini didasarkan pada besar laju pengiriman untuk 1 bit tiap detik. Seiring dengan bertambahnya jumlah paket data yang dikirim, kecepatan pengiriman juga semakin melambat. Hal tersebut dapat terjadi karena padatnya aktifitas jaringan akibat banyaknya paket data yang harus dikirim sedangkan didalam jaringan mempunyai karakteristik *bandwidth* yang berbeda.



Gambar 4.11 Perbandingan Jumlah Paket Data dengan *Bit Rate*

Pada Gambar 4.12, variasi pengiriman paket data juga berpengaruh terhadap tingkat keberhasilan data yang dikirimkan atau *throughput*. Parameter ini dapat dilihat dari selisih jumlah paket data yang dikirimkan oleh *server* dengan jumlah paket data yang diterima oleh *client*.



Gambar 4.12 Perbandingan Jumlah Paket Data dengan *Throughput*

Pada pengujian tersebut didapatkan bahwa data yang dikirimkan seluruhnya diterima oleh *client* artinya bahwa dalam jaringan tersebut data yang dikirimkan tidak ada yang hilang.

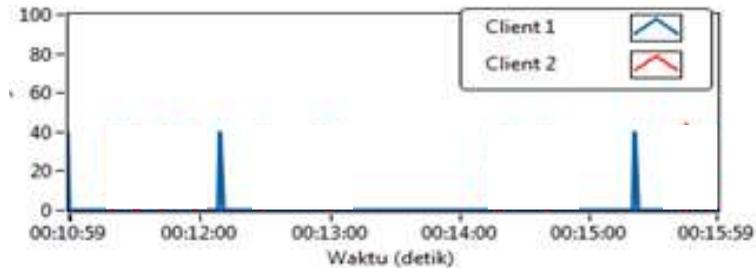
4.4.3 Respon Sistem Pengaturan Level

Pengujian dilakukan untuk mengetahui performa jaringan pada sistem pengaturan *level*. Performa jaringan yang diamati meliputi waktu tunda dan kecepatan pengiriman data. Sistem diberikat *set point* sebesar 1 meter lalu mengamati aktifitas komunikasi antara *server* dengan *client*.

Tabel 4.4 Performa Jaringan Sistem Pengaturan Level

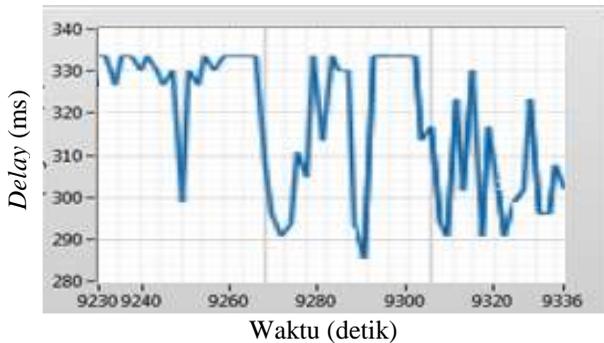
<i>Message size (bytes)</i>	72
Waktu tunda (milidetik)	40
<i>Bandwidth (kbit/detik)</i>	310
<i>Throughput (%)</i>	55,8
<i>Jitter (detik)</i>	0.001

Berdasarkan Tabel 4.4, didapatkan hasil pengiriman data sebanyak 784 paket data dengan *message size* sebesar 72 bytes sedangkan laju pengiriman data antara *server* dengan *client* sebesar 310 kbits/s. Pada Gambar 4.13 menunjukkan bahwa untuk *delay* pengiriman data muncul pada berkisar 40 milidetik.



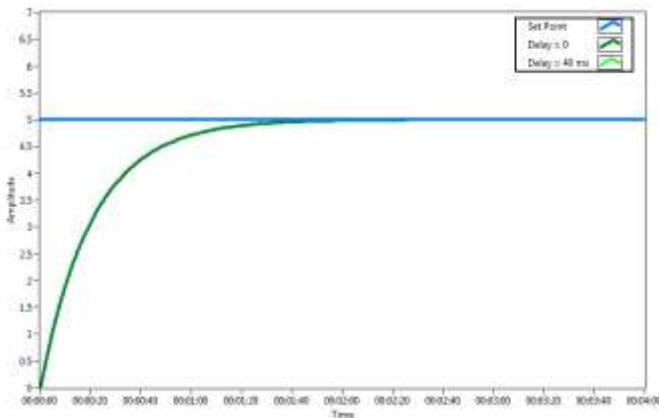
Gambar 4.13 Delay Pengiriman Data untuk Pengaturan Level

Adanya *delay* terjadi karena data yang dikirimkan membutuhkan tahapan untuk memproses sinyal menjadi data digital (waktu *pre-processing*), waktu tunggu, waktu untuk melakukan pengiriman data dari satu komputer ke komputer lain, dan waktu *post-processing* seperti pada Gambar 2.17.



Gambar 4.14 Kecepatan Pengiriman Data untuk Pengaturan *Level*

Berdasarkan Gambar 4.14 ditunjukkan kecepatan pengiriman data dari *server* ke *client*. Kecepatan pengiriman menunjukkan angka rata-rata 320 kbits/detik. Data yang dikirim dalam pengaturan *level* sebesar 72 bytes. Besar data tersebut masih dalam *range* minimal dalam *message size frame* sehingga dapat disimpulkan bahwa laju pengiriman data dalam kategori cepat.



Gambar 4.15 Pengaruh Waktu Tunda 40 milidetik terhadap Kontroler PI pada Pengaturan *Level*

Berdasarkan Gambar 4.15, *level* yang diinginkan terisi pada tangki adalah 5 meter. *Delay* awal sistem sebesar 0 detik. Adanya pengaruh

delay sebesar 40 milidetik pada pengaturan *level* mengakibatkan respon dapat mencapai keadaan tunak pada waktu 2 menit.

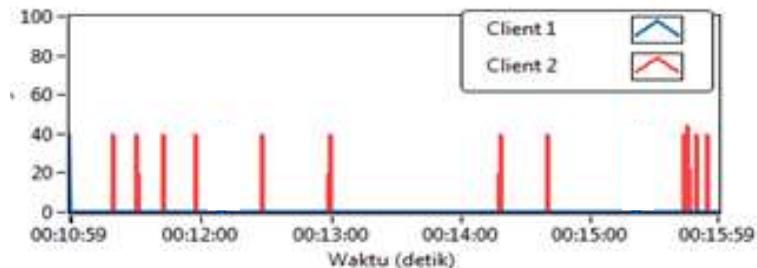
4.4.4 Respon Sistem Pengaturan Temperatur

Pengujian dilakukan untuk mengetahui performa jaringan pada sistem pengaturan *level*. Performa jaringan yang diamati meliputi waktu tunda dan kecepatan pengiriman data. Sistem diberikat *set point* sebesar 1 meter lalu mengamati aktifitas komunikasi antara *server* dengan *client*.

Tabel 4.5 Performa Jaringan Sistem Pengaturan Temperatur

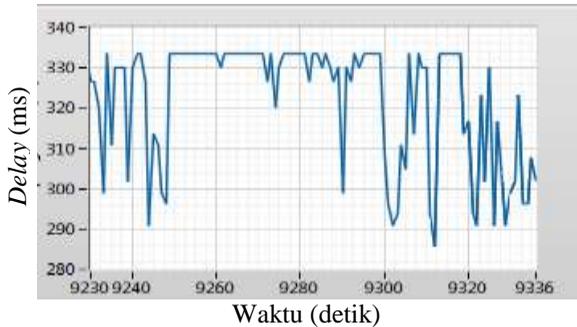
<i>Message size (bytes)</i>	72
Waktu tunda (milidetik)	40
<i>Bandwidth (kbit/detik)</i>	320
<i>Throughput (%)</i>	55,8
<i>Jitter (detik)</i>	0.001

Berdasarkan Tabel 4.4, didapatkan hasil pengiriman data sebanyak 784 paket data dengan *message size* sebesar 72 bytes sedangkan laju pengiriman data antara *server* dengan *client* sebesar 320 kbits/s. Pada Gambar 4.13 menunjukkan bahwa untuk *delay* pengiriman data muncul pada berkisar 40 milidetik.



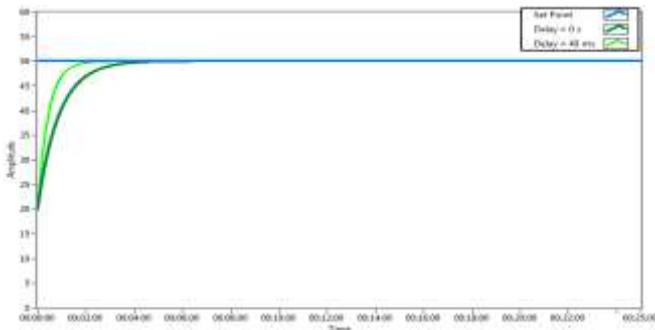
Gambar 4.16 Delay Pengiriman Data untuk Pengaturan Temperatur

Adanya *delay* terjadi karena data yang dikirimkan membutuhkan tahapan untuk memproses sinyal menjadi data digital (waktu *pre-processing*), waktu tunggu, waktu untuk melakukan pengiriman data dari satu komputer ke komputer lain, dan waktu *post-processing* seperti pada Gambar 2.17.



Gambar 4.17 Kecepatan Pengiriman Data untuk Pengaturan Temperatur

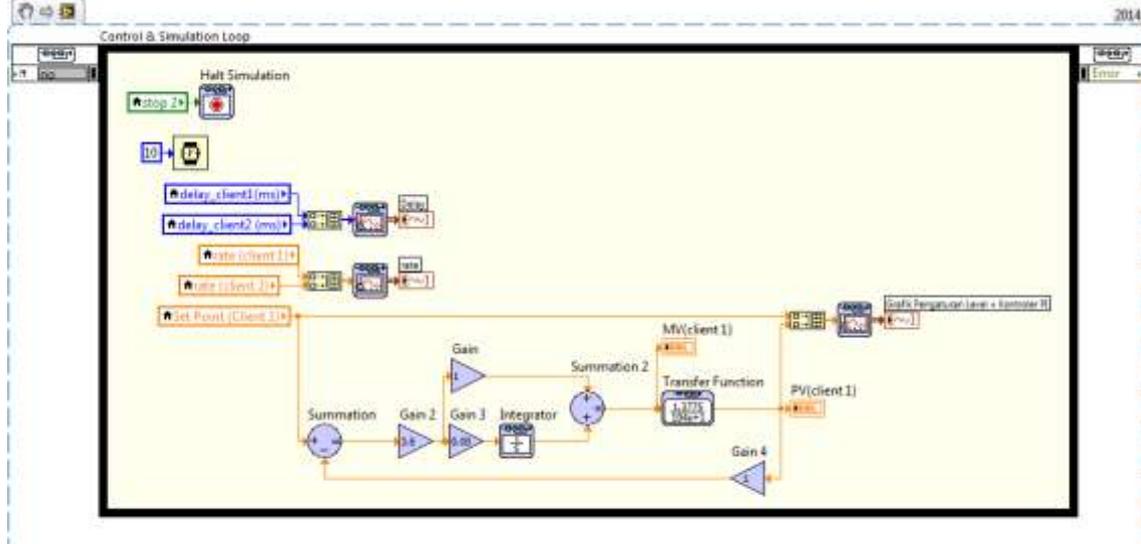
Berdasarkan Gambar 4.14 ditunjukkan kecepatan pengiriman data dari *server* ke *client*. Kecepatan pengiriman menunjukkan angka rata-rata 320 kbits/detik. Data yang dikirim dalam pengaturan *level* sebesar 72 *bytes*. Besar data tersebut masih dalam *range* minimal dalam *message size frame* sehingga dapat disimpulkan bahwa laju pengiriman data dalam kategori cepat.



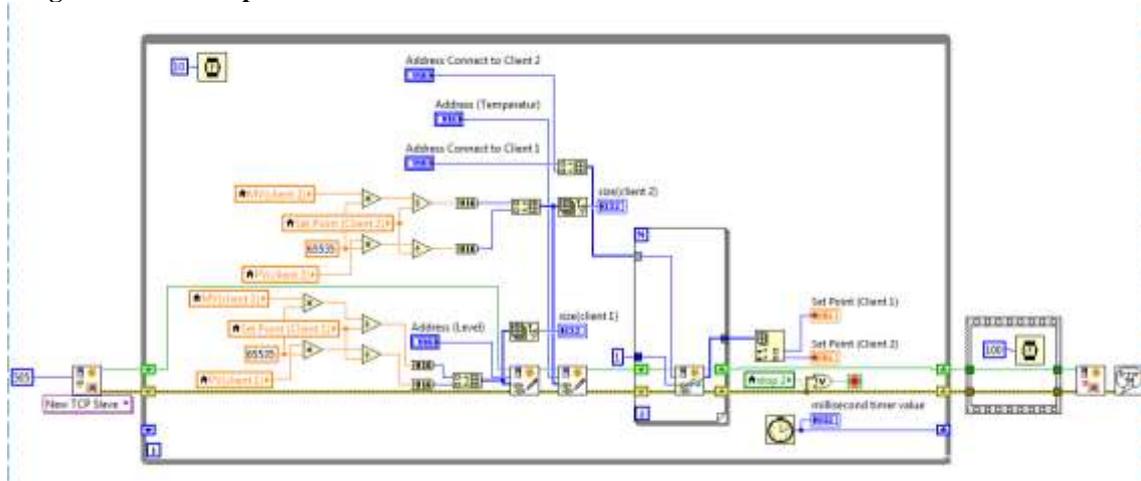
Gambar 4.18 Pengaruh Waktu Tunda 40 milidetik terhadap Kontroler PI pada Pengaturan Temperatur

Berdasarkan Gambar 4.18, temperatur yang diinginkan pada tangki berada pada kondisi 50 derajat. *Delay* awal sistem sebesar 0 detik. Adanya pengaruh *delay* sebesar 40 milidetik mengakibatkan respon mencapai keadaan tunak setelah mencapai waktu 2 menit 30 detik.

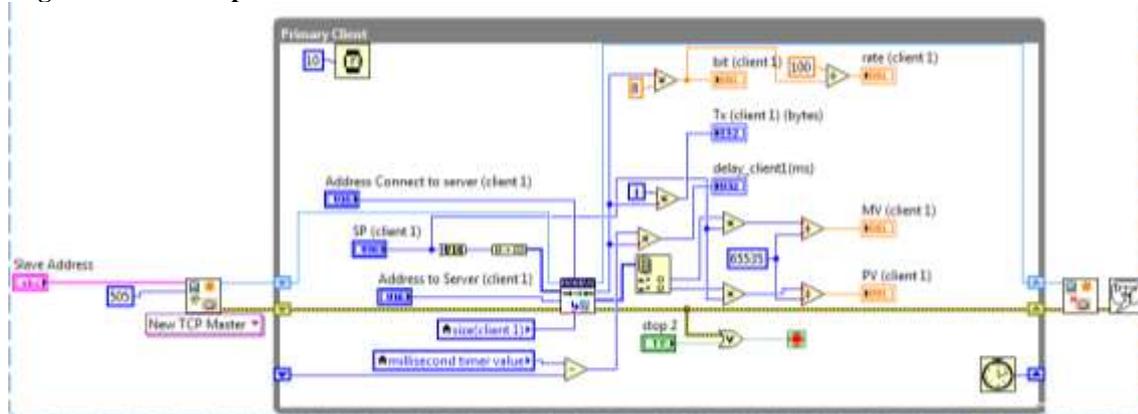
2. Program Pengaturan *Level* (Kontroler PI)



3. Program untuk komputer *Slave*



4. Program untuk komputer *Master*



BAB V

PENUTUP

5.1 Kesimpulan

Sistem pengaturan *level* dan temperatur pada tangki CPO dibangun agar menjaga *level* dan temperatur berada pada kondisi yang diinginkan. Kontroler PI digunakan karena dapat mempercepat respon dan menghilangkan kesalahan waktu tunak. Pada sistem pengaturan *level* awalnya memiliki E_{ss} sebesar 45% dapat dihilangkan terbukti dengan menggunakan kontroler PI yang mempercepat *rise time*, memperlambat *settling time* dan menghilangkan E_{ss} . Sementara pada sistem pengaturan temperatur memiliki E_{ss} sebesar 20% dan waktu respon 8 menit dapat dihilangkan menjadi 0% dan dipercepat menjadi 3 menit terbukti dengan menggunakan kontroler PI.

Penggunaan Modbus dapat menjadi protokol yang dapat menghubungkan dua sampai lebih perangkat berbasis *server* dan *client*. Hasil uji pada Modbus/TCP didapatkan performa jaringan dan perbedaan *message size* dalam pengiriman data. Perbedaan tersebut menyebabkan muncul variasi nilai *delay* pada setiap ukuran data tertentu yang menyebabkan sistem mengalami penundaan respon sistem menjadi lebih lambat. Pengujian terhadap performa jaringan juga dilakukan seperti *bandwidth*, *jitter*, *delay*, dan *throughput*. Hasilnya didapatkan bahwa respon sistem pengaturan *level* dan temperatur tidak mengalami *overshoot* dan performa jaringan komunikasi pada tangki CPO masih dalam keadaan baik dengan *delay* yang memenuhi standar sebesar 40 ms.

DAFTAR PUSTAKA

- [1] Neumann, Peter. "Communication in industrial automation-What is going on?". *Institut f. Automation und Kommunikation Magdeburg (ifak)*. Steinfeldstr. 3, 39179 Barleben. Germany. November 2007
- [2] Shahzad, A., S. Musa, A. Aborujilah and M. Irfan. "The SCADA Review: System Components, Architecture, Protocols, and Future Security Trends". *American Journal of Applied Sciences*. Vol 11 (8): 1418-1425, 2014
- [3] Federico Tramarin, Stefano Vitturi. "The IEEE 802.11n wireless LAN for real-time industrial communication". Institute of Electronics, Computer and Telecommunication Engineering. 2015.
- [4] ... "Spesifikasi Teknik Pembangunan Tank Farm. PT. MELTECH Consultindo Nusa". 2014
- [5] ... "Design Recommendation for Storage Tanks and Their Supports with Emphasis on Seismic Design". *Architectural Institute of Japan*. 2010 Edition.
- [6] Bela. G. Liptak. "*Process Software and Digital Networks*". Fourth Edition. CRC Press. 2002
- [7] Moyne, James R., M. Tilbury, Dawn. "The Emergence of Industrial Control Networks for Manufacturing Control Diagnostics, and Safety Data". *Proceedings of the IEEE*. Vol. 95, No. 1, January 2007.
- [8] Lian, Feng-Li, Moyne, James, and Tilbury, Dawn. "Network Design Consideration for Distributed Control Systems". *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 2, March 2002
- [9] ... "*Introduction to Modbus TCP/IP*". Acromag Incorporated. Wixom, MI 48393-7037 U.S.A. 2005.

- [10] “IEEE Std 802.3”. *IEEE Standard for Ethernet*. New York, NY 10016-5997. USA. 2013.
- [11] Lian, Feng-Li, Moyne, James, and Tilbury, Dawn. “Performance Evaluation of Control Network : Ethernet, ControlNet, and DeviceNet”. *IEEE Control Systems Magazine*. Volume 21
- [12] Kastner, Wolfgang. “*Advantage of Industrial Ethernet-Comparison of Modbus over TCP/IP and PROFINET*”. Vol 2. 2010.

RIWAYAT PENULIS



Eko Arisana Ismawan, lahir di Surabaya, 14 Agustus 1993 merupakan anak pertama dari dua bersaudara. Penulis menamatkan pendidikan sekolah dasar di SDN Manukan Kulon IV Surabaya, kemudian SMP Negeri 1 Sidoarjo, dan SMA Negeri 1 Sidoarjo. Setelah itu menempuh pendidikan perguruan tinggi di Politeknik Elektronika Negeri Surabaya (PENS) dengan mengambil program Studi D3 Teknik Elektronika Penulis memutuskan untuk kembali melanjutkan pendidikan S1 Teknik Elektro di Institut Teknologi Sepuluh Nopember Surabaya (ITS) dengan mengambil program studi Teknik Sistem Pengaturan. Pada tanggal 23 Juni 2016, penulis mengikuti ujian Tugas Akhir dan memperoleh gelar Sarjana Teknik