

Rancang Bangun Aplikasi Dinamis SCADA Dilengkapi Dengan Multi Koneksi

Hananta Adhi Rahmansyach¹, Basuki Rachmat², Agus Khumaidi³

^{1,2,3}Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal

Politeknik Perkapalan Negeri Surabaya

Jalan Teknik Kimia, Kampus ITS Sukolilo, Surabaya 60111

*E-mail : hanantaadhi@student.ppns.ac.id

Abstrak

Kemajuan teknologi informasi telah banyak diimplementasikan dalam bidang dunia industry 4.0. Dalam perkembangannya, peran dari sistem otomasi menjadi sangat penting. Penggunaan otomatisasi serta monitoring data pada industri bertujuan untuk membantu serta meningkatkan efektivitas pekerjaan manusia. Kebeutuhan akan *software* yang mudah, serta dapat digunakan pada banyak protokol komunikasi menjadi sangat tinggi dikarenakan kebutuhan akan pengawasan, sistem kontrol, serta akuisisi data pada industry. Agar controller dapat terhubung serta mengakuisisi data, dibutuhkan sebuah aplikasi dinamis yang berfungsi untuk menjembatani status pada plan dengan manusia. Selain itu, dibutuhkan sebuah Antarmuka Pengguna Grafis (GUI) untuk menampilkan data serta mengkonfigurasi jaringan hubung multi koneksi antara perangkat keras serta perangkat lunak sehingga memudahkan pengguna untuk pengawasan, sistem kontrol serta akuisisi data pada industri. Pada Tugas Akhir kali ini penulis menggunakan protokol komunikasi Modbus TCP/IP, Modbus RTU serta Protokol komunikasi MQTT sebagai protokol masukan. MQTT mengutamakan efisiensi data serta ukuran yang sangat ringan sehingga mendukung sistem IoT (*Internet of Things*) dan sangat tepat untuk diimplementasikan pada plan industri guna pengawasan dari jarak jauh. Pada sektor basis data penulis menggunakan basis data MySQL, Firebase serta CSV sebagai masukan. Selain itu terdapat Antarmuka Pengguna Grafis untuk mendesain HMI yang dapat digunakan sebagai tampilan antarmuka (interface) guna menunjang sistem pada industri maupun sebagai modul atau bahan ajar, khususnya pada bidang Teknik Otomasi.

Kata Kunci : Aplikasi Dinamis, Modbus, IoT, GUI, MQTT

1. PENDAHULUAN

Teknologi informasi telah banyak diterapkan di dunia industri. Penggunaan sistem kendali otomatisasi dalam industri bertujuan untuk membantu dan meningkatkan efektifitas kerja manusia. Saat ini kita berada di tengah periode Industri 4.0 dimana angka 4.0 merepresentasikan Revolusi Industri yang telah memasuki era gelombang ke-4. Ciri-ciri model dari Industri 4.0 merupakan gabungan dari beberapa perkembangan teknologi terkini yang meliputi teknologi informasi dan komunikasi, jaringan komunikasi, big data dan cloud computing serta peralatan yang dikembangkan untuk memudahkan interaksi manusia dengan komputer; [1]. Semua aspek komputasi industri dapat saling berkomunikasi dengan memanfaatkan protokol komunikasi, salah satunya ialah protokol komunikasi Modbus.

Protokol Komunikasi adalah bahasa untuk saling berkomunikasi antara perangkat keras, dan perangkat lunak. Modbus adalah metode

Protokol komunikasi dua serial, di mana komunikasi dapat terjadi dengan sistem *Master and Slave* pada Modbus; [2]. Modbus dapat ditransmisikan melalui komunikasi serial (Modbus RTU), melalui komunikasi Ethernet (Modbus TCP/IP), melalui ASCII (Modbus ASCII) dan modbus plus. Modbus RTU juga dilengkapi dengan mekanisme cyclic redundancy error (CRC) yang berfungsi untuk mengurangi kesalahan dan memastikan keandalan data. Dalam prakteknya, Modbus mendukung banyak modem dan gateway untuk bekerja; [3]

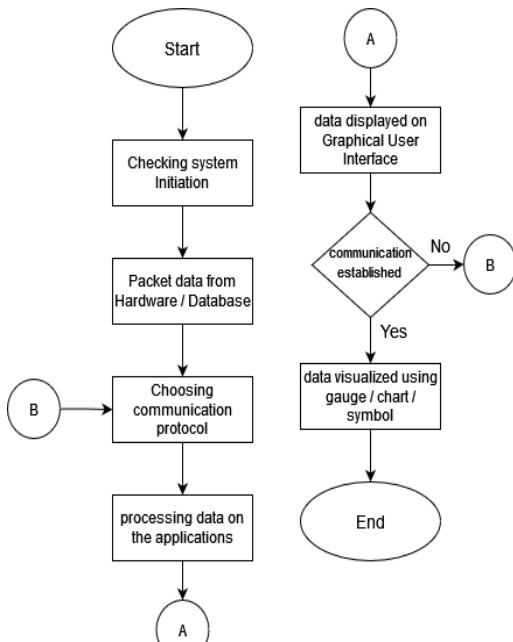
2. METODE

2.1 Metode Penelitian

Perancangan sistem awal dibuat dalam bentuk diagram alur seperti yang dapat dijelaskan pada **Gambar 2.1**

Berdasarkan diagram flowchart, perancangan sistem merupakan panduan dalam perancangan kerangka sistem dan perangkat lunak. Untuk memastikan seluruh sistem yang telah dirancang berjalan sesuai dengan yang

diharapkan maka dilakukan pengujian sistem. Setelah pengujian sistem berhasil, tahap selanjutnya adalah pembuatan buku laporan.



Gambar 2.1 Research Flowchart

2.2 Identifikasi Masalah

Pada sistem ini permasalahan yang diangkat adalah upaya untuk dapat merancang aplikasi dinamis dengan multi koneksi serta antarmuka pengguna grafis untuk membuat *Human Machine Interface* (HMI) berbasis *Graphical User Interface* (GUI). Dengan upaya penerapan dari protokol Modbus TCP/IP, Modbus RTU dan MQTT untuk komunikasi data modul dengan aplikasi serta mengintegrasikan database MySQL, Firebase dan CSV untuk transaksi data dari server ke aplikasi.

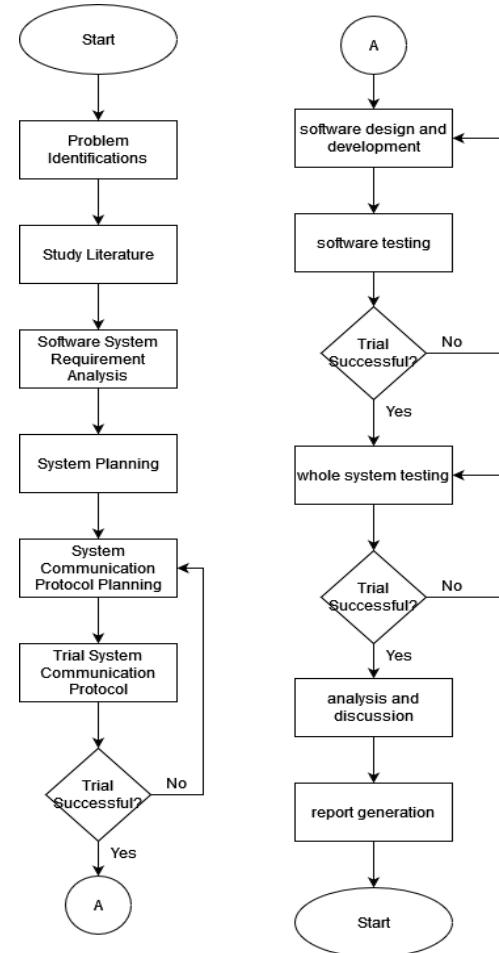
2.3 Studi Literatur

Pada tahap ini penulis mencari informasi sebanyak-banyaknya tentang konsep-konsep yang akan digunakan dalam penelitian. Pencarian dilakukan terkait informasi tentang Protokol Komunikasi Modbus; [4], Protokol MQTT; [5] serta MySQL Database; [6].

2.4 Perencanaan Sistem

Sistem aplikasi dinamis berjalan pada kerangka ASP .NET MVC. Didukung oleh Modbus, protokol MQTT dan beberapa DBMS untuk transaksi data, antara lain MySQL, Firebase dan CSV. penyajian data pada objek seperti pengukur atau grafik dilakukan dengan menghubungkan nama variabel pada tag

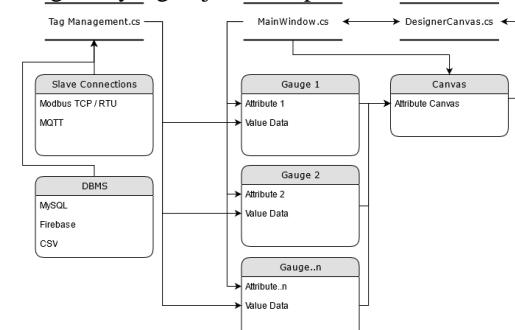
manager. Flowchart sistem pada penelitian ini ditunjukkan pada **Gambar 2.2**



Gambar 2.2 System Flowchart

2.5 Object Oriented Programming

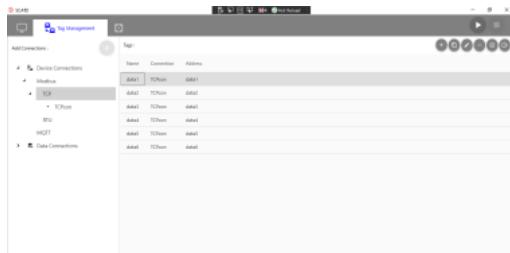
Dalam aplikasi dinamis ini, penyusunan kerangka sistem menggunakan pemrograman berorientasi objek sebagai dasar penyusunan struktur program. Langkah-langkah yang dilakukan untuk mendeteksi objek seperti pada diagram yang dijelaskan pada **Gambar 2.3**.



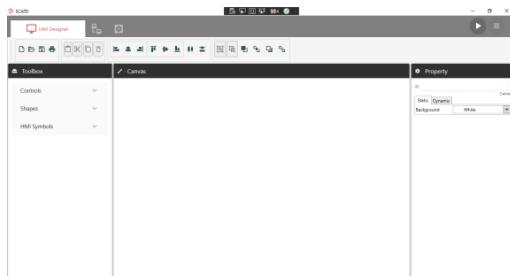
Gambar 2.3 Object Oriented Programming

2.6 Desain Perangkat Lunak

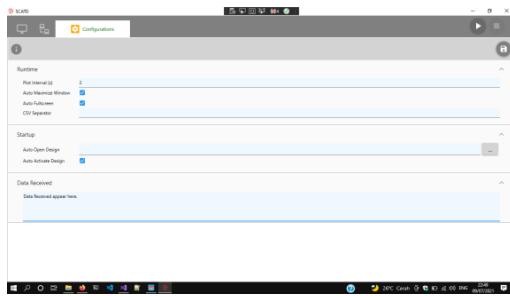
Aplikasi SCATO memiliki tiga workspace, diantaranya adalah tab Tag Management (A), Graphic Designer (B) dan Configurations (C) dimana masing-masing tab workspace memiliki fungsi yang berbeda seperti terlihat pada **Gambar 2.4**.



(A)



(B)



(C)

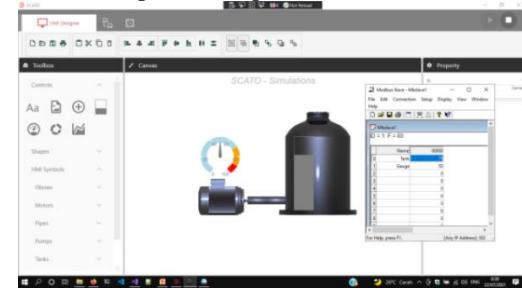
Gambar 2.4 Perangkat Lunak SCATO

3. HASIL DAN PEMBAHASAN

3.1 Protokol Komunikasi Modbus

Aplikasi dinamis SCATO menggunakan beberapa protokol komunikasi Modbus, termasuk Modbus TCP/IP dan protokol Modbus RTU. Untuk berkomunikasi satu sama lain, data diatur menggunakan nomor registri untuk setiap komunikasi Modbus. Setiap perangkat atau perangkat yang saling terhubung menggunakan Modbus membutuhkan akses Modbus sebagai protokol komunikasi yang sesuai dengan setiap nomor registri pada akses Modbus. pada Modbus TCP parameter yang digunakan adalah IP Address dan juga di sisi lain parameter komunikasi

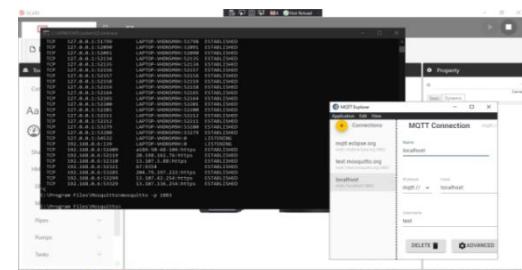
Modbus RTU yang digunakan adalah serial port dan baudrate. Koneksi sukses yang telah dibuat menggunakan komunikasi protokol Modbus dapat dilihat pada **Gambar 3.1**.



Gambar 3.1 Modbus Protocol

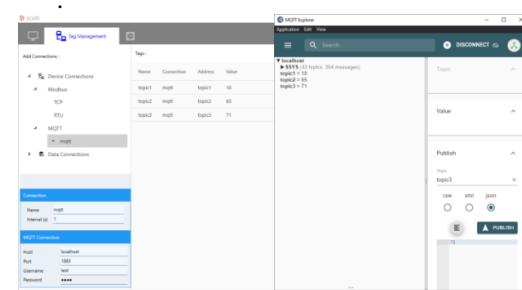
3.2 Protokol Komunikasi MQTT

Sistem kerja Protokol Komunikasi MQTT menggunakan metode data *Publish and Subscribe*. Dalam penerapannya, software SCATO akan terhubung ke broker dan memiliki topik tertentu, yang dalam studi kasus ini broker yang digunakan adalah 'mosquitto' dengan konfigurasi lokal dan bantuan aplikasi MQTT Explorer sebagai interface.



Gambar 3.2 Mosquitto dan MQTT Explorer

Setelah mengatur konfigurasi untuk MQTT, data yang diterima akan secara otomatis dipisahkan dan dijelaskan berdasarkan topic yang disubscribe

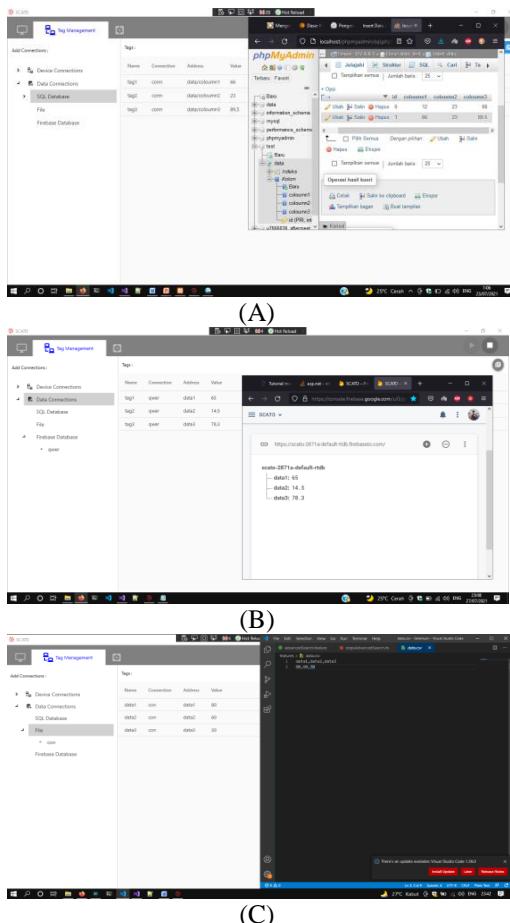


Gambar 3.3 Protokol MQTT

3.3 Manajemen Basis Data

Beberapa protokol koneksi database yang didukung oleh SCATO diantaranya ialah MySQL, Firebase dan CSV dengan tipe data int, float, double dan boolean. Data akan

diterima dan diurai berdasarkan setiap kolom pada tabel database secara otomatis. Komunikasi dengan database dapat dilakukan dengan bantuan konektor berupa framework library seperti SQL Connections dan Firesharp yang tersedia dalam Paket Library NuGet.



Gambar 3.4 Database Connection Protocol

3.4 Keberhasilan Tes Komunikasi

Dalam pengujian komunikasi berikut, SCATO menggunakan variabel untuk menguji koneksi antara perangkat keras dan aplikasi SCATO. Pada pengujian ini untuk membuktikan keberhasilan koneksi yang telah dibuat dimana hasilnya dapat dilihat pada **Tabel 3.1 - 3.6**

Table 3.1 Tabel Keberhasilan Modbus TCP

Variabel	Data masuk	Data Pembacaan SCATO
Data1	28	28
Data2	60	60
Data3	0	0

Data1	28	28
Data2	60	60
Data3	0	0
Data1	29	29
Data2	60	60
Data3	1	1
Data1	28	28
Data2	59	59
Data3	1	1
Data1	29	29
Data2	60	60
Data3	1	1

Table 3.2 Tabel Keberhasilan Modbus TCP

Variabel	Data Masuk	Data Pembacaan SCATO
Data1	28	28
Data1	28	28
Data1	28	28
Data1	29	29
Data1	29	29
Data1	29	29
Data1	28	28
Data1	28	28
Data1	29	29

Table 3.3 Tabel Keberhasilan MQTT

Variabel	Data Masuk	Data Pembacaan SCATO
Topic1	28	28
Topic2	60	60
Topic3	0	0
Topic1	28	28
Topic2	60	60
Topic3	0	0

Topic1	29	29
Topic2	59	59
Topic3	0	0
Topic1	27	27
Topic2	59	59
Topic3	0	0
Topic1	29	29
Topic2	60	60
Topic3	0	0

Table 3.4 Tabel Keberhasilan DBMS MySQL

Variabel	Data Masuk	Data Pembacaan SCATO
Tag1	67	67
Tag2	22	22
Tag3	96,5	96,5
Tag1	12	12
Tag2	76	76
Tag3	33,8	33,8
Tag1	60	60
Tag2	12	12
Tag3	32,1	32,1
Tag1	88	88
Tag2	87	87
Tag3	66,5	66,5
Tag1	24	24
Tag2	55	55
Tag3	11,8	11,8

Table 3.5 Tabel Keberhasilan DBMS Firebase

Variabel	Data Masuk	Data Pembacaan SCATO
Data1	65	65
Data2	14,5	14,5

Data3	78,3	78,3
Data1	11	11
Data2	22,8	22,8
Data3	54,1	54,1
Data1	79	79
Data2	60,8	60,8
Data3	87,5	87,5
Data1	05	05
Data2	4,11	4,11
Data3	32,9	32,9
Data1	42	42
Data2	11,2	11,2
Data3	32,5	32,5

Table 3.6 Tabel Keberhasilan File CSV

Variabel	Data Masuk	Data Pembacaan SCATO
Data1	80	80
Data2	60	60
Data3	30	30
Data1	12	12
Data2	66	66
Data3	43	43
Data1	91	91
Data2	28	28
Data3	32	32
Data1	72	72
Data2	94	94
Data3	76	76
Data1	65	65
Data2	22	22
Data3	50	50

3.5 Kecepatan Pembacaan Data

kecepatan pembacaan data sangat berpengaruh terhadap kinerja aplikasi dimana

aplikasi untuk pengawasan (*monitoring*) harus menampilkan data secara real time. Semakin kecil latency atau delay, semakin baik aplikasi menampilkan data secara otomatis. Hasil tes membaca dapat dilihat pada **Tabel 3.7-3.12.**

Tabel 3.7 Kecepatan Pembacaan Data Modbus TCP

Variabel	Kecepatan Pembacaan data (s)
Data1	0.52
Data2	0.61
Data3	0.51

Tabel 3.8 Kecepatan Pembacaan Data Modbus RTU

Variabel	Kecepatan Pembacaan Data (s)
Data1 (Percobaan ke-1)	0.61
Data1 (Percobaan ke-2)	0.58
Data1 (Percobaan ke-3)	0.64

Tabel 3.9 Kecepatan Pembacaan Data MQTT

Variabel	Kecepatan Pembacaan data (s)
Topic 1	0.56
Topic 2	0.64
Topic 3	0.47

Tabel 3.10 Kecepatan Pembacaan Data MySQL

Variabel	Kecepatan Pembacaan Data (s)
Tag1	0.48
Tag2	0.48

Tag3	0.56
------	------

Tabel 3.11 Kecepatan Pembacaan Data Firebase

Variabel	Kecepatan Pembacaan Data (s)
Data1	0.56
Data2	0.56
Data3	0.40

Tabel 3.12 Kecepatan Pembacaan Data CSV

Variabel	Kecepatan Pembacaan Data (s)
Data1	0.42
Data2	0.56
Data3	0.62

4. PENUTUP

4.1 Kesimpulan

Berdasarkan pengujian serta analisa sistem yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Komunikasi antara Modbus TCP-IP serta Modbus RTU dengan aplikasi SCATO berjalan dengan tingkat akurasi pembacaan data sebesar 100 % serta kecepatan pembacaan data sebesar 0,55 detik dan 0,61 detik.
2. Komunikasi antara MQTT dengan Aplikasi SCATO berjalan dengan tingkat akurasi pembacaan data sebesar 100 % serta kecepatan pembacaan data sebesar 0,55 detik.
3. Komunikasi antara database MySQL, Firebase serta CSV dapat berjalan dengan tingkat akurasi pembacaan data sebesar 100 % serta masing-masing database memiliki kecepatan pembacaan data sebesar 0,5 detik, 0,5 detik dan 0,53 detik.

4.2 Saran

Pada tugas akhir kali ini, aplikasi dinamis SCATO masih belum bekerja secara sempurna. Terdapat beberapa kekurangan serta keterbatasan, oleh karena itu diperlukan adanya pengembangan untuk penelitian selanjutnya dimana:

1. Menambahkan fungsi umpan balik (feedback) dimana SCATO tidak hanya

- dapat mengambil data atau memonitoring plan namun juga dapat mengontrol plant dengan masukan berupa tombol ataupun *input* numeric.
2. Menambahkan beberapa protokol komunikasi yang umum dijumpai pada industri seperti IEC Standar, DNP 3.0, PROFIBUS, OPC serta INDACTIC.

5. DAFTAR PUSTAKA

- [1] R. Fauzan, "Karakteristik Model dan Analisa Peluang-Tantangan Industri 4.0," *J. Tek. Inform. Politek. Hasnur*, vol. 4, no. 1, pp. 1–11, 2018, [Online]. Available: <http://ejournal.polihasnur.ac.id/index.php/pha/article/view/271>.
- [2] Nurpadmi, "Tudi Tentang Modbus Protokol Pada Sistem Kontrol," *Forum Teknol.*, vol. 01, no. 2, 2010.
- [3] S. D. Chandra, H. Kusuma, and Suwito, "Desain Dan Implementasi Protokol Modbus Untuk Sistem Antrian Terintegrasi Pada Pelayanan Surat Izin Mengemudi (Sim) Di Kepolisian Resort," 2016.
- [4] K. Wang, D. Peng, L. Song, and H. Zhang, "Implementation of Modbus communication protocol based on ARM Coretx-M0," *Conf. Proc. - 2014 Int. Conf. Syst. Sci. Eng. ICSSE 2014*, pp. 69–73, 2014, doi: 10.1109/ICSSE.2014.6887907.
- [5] R. A. Atmoko and D. Yang, "Online Monitoring & Controlling Industrial Arm Robot Using MQTT Protocol," *Proc. 2018 Int. Conf. Robot. Biomimetics, Intell. Comput. Syst. Robionetics 2018*, pp. 12–16, 2019, doi: 10.1109/ROBIONETICS.2018.8674672.
- [6] H. Gorskis, "SQL query construction from database concepts," *59th Int. Sci. Conf. Inf. Technol. Manag. Sci. Riga Tech. Univ. ITMS 2018 - Proc.*, pp. 1–4, 2018, doi: 10.1109/ITMS.2018.8552953.