

Rancang Bangun Internet of Things (IoT) pada Pengendalian Tegangan Simulator Input Output Berbasis PLC-ESP32

Fajriati Amalia Putri¹, Sarjono Wahyu Jadmiko², Sofian Yahya³

¹Jurusan Teknik Elektro, Politeknik Negeri Bandung, Bandung 40012

E-mail : fajriati.amalia.toi17@polban.ac.id

²Jurusan Teknik Elektro, Politeknik Negeri Bandung, Bandung 40012

E-mail : sarjono_wahyu@polban.ac.id

³Jurusan Teknik Elektro, Politeknik Negeri Bandung, Bandung 40012

E-mail : softanyahya@polban.ac.id

ABSTRAK

Sistem IoT saat ini masih umum menggunakan mikrokontroler sedangkan pada bidang industri jenis kontroler yang sering digunakan adalah PLC. Lalu pada pengendalian tegangan simulator plant berbasis PLC umumnya tidak dapat di kontrol secara IoT melalui platform IoT, sedangkan perangkat pada platform IoT kini menjadi salah satu alternatif yang dipilih oleh industri guna untuk mengefisienkan dalam penyimpanan data serta dalam pengontrolan maupun pengendalian sebuah sistem dengan respon waktu yang sangat cepat (tanpa *delay*). Oleh karena itu pada penelitian ini penulis membuat sebuah pengendalian tegangan analog input menggunakan simulator input output berbasis PLC IoT. PLC yang digunakan pada sistem ini yaitu PLC CP1H dengan *ubidots* sebagai platform IoT untuk menyimpan data, menampilkan grafik, serta melakukan pengendalian dan pengontrolan sistem dengan memanfaatkan ESP32 sebagai perantaranya. Berdasarkan hasil pengujian, pada analog input tegangan yang diukur 0-10 Volt dengan *sampling data* 100 mili detik, *Ubidots* dapat menampilkan data secara numerik dan grafik dengan *delay* rata-rata 1 detik.

Kata Kunci

ESP32, Simulator plant, PLC, Ubidots, IoT

1. PENDAHULUAN

Internet of things kini menjadi industri yang paling cepat tumbuh. Platform IoT menawarkan beberapa layanan yang mempermudah pengembangan project dan mengatur perangkat untuk secara jarak jauh mengelola perangkat. Singkat kata, platform IoT adalah teknologi *multilayer* yang memungkinkan pengguna untuk mengelola perangkat-perangkat yang terhubung, sementara secara bersamaan menangani masalah skalabilitas hingga keamanan perangkat. Sistem IoT saat ini masih umum menggunakan mikrokontroler sedangkan pada bidang industri jenis kontroler yang sering digunakan adalah PLC.

Platform IoT memainkan peran penting dalam arsitektur IoT. Dimana ketika seorang developer membangun sebuah project IoT atau sistem yang berbasis sensor dan perangkat yang mengirimkan data ke platform cloud. Platform ini akan menyimpan data dan menggunakannya untuk membentuk sebuah grafik. Ubidots adalah salah satu platform IoT yang menyediakan beragam layanan.

Sebelum berkembangnya sistem IoT proses pemantauan serta pengontrolan biasa dilakukan pada personal komputer. Agar bisa berkomunikasi antara PLC dan personal komputer digunakan protokol komunikasi modbus serial RS485, hasil dari penelitian tersebut respon yang dibutuhkan pada sistem komunikasi ini rata-rata 2,6 detik dan jarak sambungan kabel maksimal 100 meter [1]. Seiring dengan adanya perkembangan

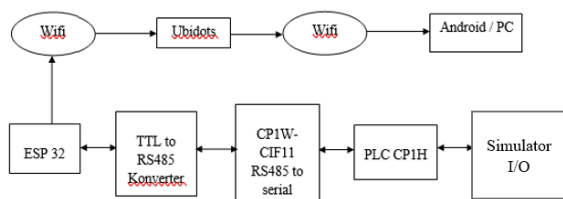
teknologi, sistem IoT mulai dikenal dan diterapkan pada suatu sistem kontrol berbasis mikrokontroler. Dengan NodeMCU sebagai kontroler dan ThingsSpeak sebagai web server, beberapa dari peneliti berhasil membuat sebuah sistem IoT yang dapat dimonitor melalui smartphone android. Sistem IoT ini digunakan juga pada kendali smarthome, namun waktu respon yang dihasilkan masih terlalu lama [2]. Sistem IoT juga dapat diterapkan sebagai pemantau untuk mesin penetas telur dengan menggunakan kontroler yang sama yaitu NodeMCU mampu menghasilkan respon waktu yang lebih cepat yaitu 18,84 detik untuk melakukan pengiriman serta penerimaan data, namun terdapat kekurangan pada error pembacaan yang mencapai 13.33% [3]. Disisi lain peneliti juga melakukan pengembangan pada sistem IoT yang diterapkan melalui PLC. Diaplikasikan pada sistem pengangkut sampah, untuk proses pengawasan serta pengontrolan dapat dilakukan melalui pengiriman email dengan bantuan mikrokontroler didalamnya [4]. Karena terlalu banyaknya prosedur yang perlu dijalankan maka metode IoT yang diimplementasikan pada PLC diganti menggunakan modem GSM dengan komunikasi RS232. Sistem yang diaplikasikan pada purwarupa SCADA PLTMH berhasil dengan error pembacaan yang mendekati 0% dan respon waktu pengiriman data rata rata 10,38% [5]. Respon yang dibutuhkan harus lebih cepat, karena itu untuk mengatasinya dengan menggunakan PLC siemens Logo dan Raspberry Pi sebagai web server, hasilnya yaitu respon waktu menjadi 4,15 detik dengan metode pemantauan berupa notifikasi

pada aplikasi sosial media [6]. Pada perancangan miniature pengisian air otomatis menggunakan adruino nano berbasis IoT *delay* untuk sensor membaca volume air paling kecil sebesar 2 detik dengan rata-rata tegangan *output* sebesar 4.70 VDC [7]. Implementasi IoT dan monitoring pada generator set 2 KVA menggunakan Node MCU ESP32 dan *Thingspeak* sebagai web servernya, karena kontrol jarak jauh tetap membutuhkan waktu respon yang sedikit lama sehingga lebih efisien dan efektif apabila monitoring tanpa jarak [8]. Selanjutnya pada sistem kendali dan monitoring penggunaan daya listrik pada perangkat listrik rumah berbasis IoT, setelah dilakukan pengujian sebanyak 30 kali diperoleh hasil dengan tingkat *error* yang rendah sebesar 3.13% pada saklar 1 dan 3.41% [9]. Karena dirasa *error* yang diperoleh masih cukup besar, dilakukan penelitian dengan judul *prototype* sistem kontrol dan monitoring kubikel menggunakan konsep IoT berbasis mikrokontroler aduino dan hasil pembacaan sensor yang dikirim ke LCD dan web sudah menghasilkan *error* yang sangat kecil yaitu sebesar 0.94% [10].

Dari kesimpulan diatas, penulis memutuskan untuk membuat rancang bangun pengendalian tegangan simulator I/O dengan metode PID berbasis PLC yang dapat diakses melalui smartphone android dan personal komputer dengan perangkat tambahan mikrokontroler ESP32 sebagai perantara anantara PLC CPIH dan *Ubidots* sebagai web server dengan protocol komunikasi modbus. Dengan harapan keluaran tegangan dari simulator orde dua yang akan dikendalikan tidak ada jeda atau *delay*.

2. METODE PENELITIAN

2.1 Perancangan Sistem



Gambar 1. Blok diagram sistem

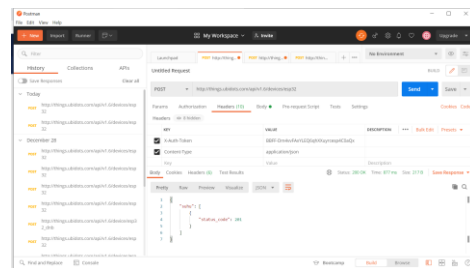
Gambar 1 memperlihatkan blok diagram dari sistem. Pada gambar diatas membahas mengenai sistem komunikasi antar ESP32 dan PLC CPIH menggunakan protokol modbus RTU RS485 yang dilengkapi oleh IoT melalui *ubidots* untuk melakukan pemantauan dan pengontrolan tegangan analog input pada simulator I/O dari jarak jauh dengan harapan komunikasi yang tanpa jeda (tidak ada *delay*).

Sistem IoT pada proses pengontrolan dan pengendalian tegangan analog input pada simulator I/O menggunakan android atau komputer sebagai HMI yang dapat diakses menggunakan *Ubidots*. Data yang dikirimkan oleh pengguna akan diterima oleh ESP32 dan meneruskan perintahnya terhadap kontroler PLC pada simulator I/O

yang telah terhubung dengan komunikasi modbus dengan ESP32.

Begitu pula untuk parameter atau data yang terdapat pada simulator I/O akan dikirimkan menuju ESP32 yang akan meneruskan data melalui web server *Ubidots* sehingga pengguna dapat memantau melalui android ataupun komputer. Pada system ini ESP32 berfungsi sebagai perantara anantara *Ubidots* dan simulator I/O yang dikontrol menggunakan PLC. Sehingga, semua data yang berasal dari simulator I/O maupun dari pengguna akan disimpan sementara pada ESP32 sebelum diteruskan menuju simulator atau pengguna.

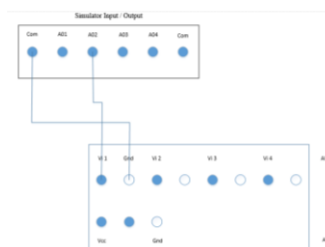
2.2 Perancangan Pada Postman



Gambar 2. Pengiriman data dari postman

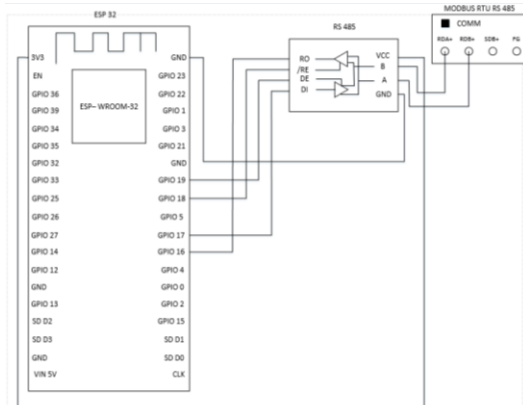
Perancangan pada POSTMAN ditujukan untuk *testing* (uji coba) REST API sehingga akan terbaca nantinya pada *ubidots*. Setelah membuka *software* Postman, aturlah *field* pada *menu* *collection* menjadi POST yang digunakan untuk mengirim data tersebut agar terbaca pada *Ubidots*. Lalu alamat yang dicantumkan pada postman yaitu <http://things.ubidots.com/api/v1.6/devices/esp32>, *devices* pada *Ubidots* harus sesuai dengan yang dibuat, disini *name devices* yang dibuat yaitu *esp32*. Pada *menu headers* ada bagian X-Auth-Token dan Content-Type. Untuk X-Auth-Token *value*-nya diisikan token sesuai *device Ubidots* yang dibuat, sedangkan pada Content-Type *value* yang dimasukkan yaitu *application/json*. Langkah selanjutnya yaitu pada *menu Body* masukkan data yang akan dikirimkan ke *Ubidots* guna untuk mengetahui apakah *device* yang dibuat pada *Ubidots* sudah benar atau belum. Pada contoh ini menggunakan suhu dan memasukkan data 50. Setelah semua Langkah dilakukan, data siap dikirimkan dengan klik *send*, dan data uji coba akan terbaca langsung pada *device di Ubidots* secara tampilan grafik ataupun numerik.

2.3 Perancangan pada ESP32 Menggunakan Simulator I/O



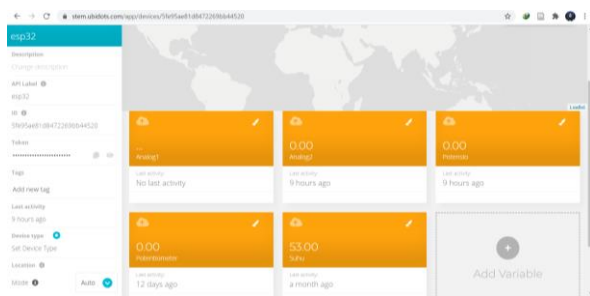
Gambar 3. Rangkaian simulator I/O dan PLC CPIH

Perancangan pada ESP32 menggunakan simulator I/O yang terhubung dengan PLC CP1H dan RS485 ditujukan untuk melihat data berupa tegangan dapat ditampilkan atau tidak pada ubidots, baik tampilan numerik ataupun grafik. Selanjutnya untuk rangkaian yang telah disambungkan ke PLC CP1H dihubungkan ke Modbus RTU Rs485 yang terhubung ke PLC, selanjutnya di hubungkan ke Rs485 dan ESP32, dan terakhir data tersebut akan dimonitoring melalui Ubidots.



Gambar 4. Rangkaian modbus RTU, RS485, dan ESP32

Pada Cx-Programmer modul Slave dipakai karena PLC CP1H akan dijadikan slave dan untuk masternya yaitu ESP32, dan disini hanya akan menggunakan satu analog input. Untuk setting pada Cx-Programmer disini diatur untuk range dari 0 s/d 10 Volt. Lalu pada Built-in analog resolution, D1 diset &6000 karena resolusi yang digunakan ialah 6000. Selanjutnya menginstall ESP32 pada Arduino IDE dan menyesuaikan board dengan ESP32 yang digunakan lalu input program agar data dapat dimonitoring pada Ubidots.

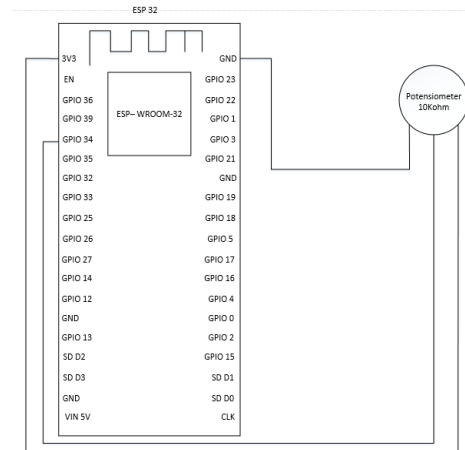


Gambar 5. Pengaturan device dan variable pada ubidots

Pada Ubidots, buatlah device dan variable sesuai dengan yang sudah dibuat pada program di Arduino IDE, dan perlu diperhatikan untuk ssid serta password wifi yang tersambung ke ESP32 tidak salah. Langkah berikutnya setelah semua device sesuai, upload program yang telah dibuat dan data akan terbaca pada Ubidots dengan cara mengatur simulator I/O dengan sampling data 100 mili detik sekali.

2.4 Perancangan Pada ESP32 Menggunakan Potensiometer

Perancangan pada ESP32 menggunakan potensiometer, tidak jauh berbeda dengan perancangan pada ESP32 menggunakan simulator I/O, hanya disini tidak menggunakan PLC CP1H dan RS485. Berikut gambar rangkaian untuk perancangan pada ESP32 menggunakan potensiometer.



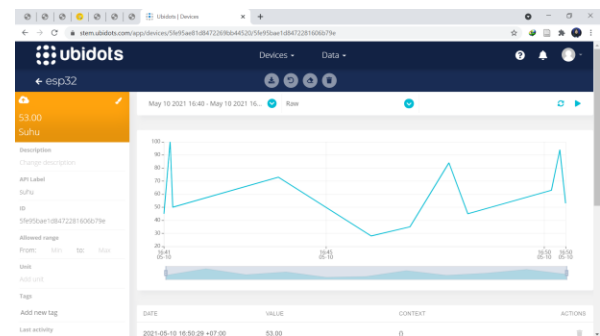
Gambar 6. Rangkaian potensiometer dan ESP32

Untuk data sama seperti sebelumnya yaitu dikirimkan ke Ubidots agar dapat dimonitoring dan dengan *sampling* data setiap 100 mili detik.

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Nilai Dari Postman Ke Ubidots

Pengujian nilai dari Postman ke Ubidots ditujukan untuk mengamati pengujian nilainya berhasil terbaca atau tidak pada Ubidots.



Gambar 7. Hasil pembacaan nilai dari postman dengan tampilan grafik

Dari beberapa *sampling* data tersebut lalu dikirimkan ke Ubidots agar dapat terbaca pada *device* Ubidots yang sudah dibuat. Tabel berikut merupakan penunjukkan data yang terkirim ke Ubidots beserta waktunya secara *real time*.

Tabel 1. Data yang diterima ubidots dari postman

No.	Nilai yang di Input	Tanggal Diterima	Waktu Diterima
1.	100	10 Mei 2021	16:41:25
2.	50	10 Mei 2021	16:41:29
3.	73	10 Mei 2021	16:43:54
4.	28	10 Mei 2021	16:46:02
5.	35	10 Mei 2021	16:46:55
6.	84	10 Mei 2021	16:47:48
7.	45	10 Mei 2021	16:48:14
8.	63	10 Mei 2021	16:50:09
9.	94	10 Mei 2021	16:50:21
10.	53	10 Mei 2021	16:50:29

Dari gambar yang telah dilampirkan, dapat disimpulkan bahwa pengujian nilai yang dibuat pada Postman dan dikirimkan ke *Ubidots* dapat terbaca secara *real time* dan dapat dilihat secara grafik ataupun numeriknya dengan beberapa *sampling* data.

3.2 Pengujian Pada ESP32 Menggunakan Simulator I/O



Gambar 8. Hasil pembacaan nilai pada ubidots dari ESP32 menggunakan simulator I/O secara grafik

Pengujian Pada ESP32 menggunakan simulator I/O yang disambungkan dengan PLC CP1H dan RS485 ditujukan untuk mengamati data dapat diterima atau tidak pada *Ubidots* serta dapat berubah sesuai pengaturan pada simulatornya. Disini *sampling* data yang diperoleh yaitu setiap 100 milidetik.

Tabel 2. Data yang diterima ubidots dari ESP32 menggunakan simulator I/O

No.	Nilai yang Terbaca	Tanggal Diterima	Waktu Diterima
1.	0.00	04 Juli 2021	11:31:01
2.	0.00	04 Juli 2021	11:31:02
3.	0.22	04 Juli 2021	11:31:03
4.	0.39	04 Juli 2021	11:31:04
5.	0.70	04 Juli 2021	11:31:06
6.	0.98	04 Juli 2021	11:31:07
7.	1.18	04 Juli 2021	11:31:08
8.	1.27	04 Juli 2021	11:31:09
9.	1.68	04 Juli 2021	11:31:11
10.	2.01	04 Juli 2021	11:31:12
11.	2.21	04 Juli 2021	11:31:13
12.	2.58	04 Juli 2021	11:31:15
13.	2.84	04 Juli 2021	11:31:16
14.	3.20	04 Juli 2021	11:31:17

15.	3.41	04 Juli 2021	11:31:18
16.	3.81	04 Juli 2021	11:31:20
17.	4.13	04 Juli 2021	11:31:21
18.	4.48	04 Juli 2021	11:31:23
19.	4.89	04 Juli 2021	11:31:24
20.	5.21	04 Juli 2021	11:31:25
21.	5.61	04 Juli 2021	11:31:27
22.	5.92	04 Juli 2021	11:31:29
23.	6.36	04 Juli 2021	11:31:31
24.	6.60	04 Juli 2021	11:31:32
25.	6.70	04 Juli 2021	11:31:34
26.	7.01	04 Juli 2021	11:31:35
27.	7.21	04 Juli 2021	11:31:36
28.	7.34	04 Juli 2021	11:31:37
29.	7.48	04 Juli 2021	11:31:38
30.	7.73	04 Juli 2021	11:31:40
31.	8.06	04 Juli 2021	11:31:42
32.	8.19	04 Juli 2021	11:31:43
33.	8.25	04 Juli 2021	11:31:45
34.	8.43	04 Juli 2021	11:31:46
35.	8.83	04 Juli 2021	11:31:47
36.	9.03	04 Juli 2021	11:31:48
37.	9.21	04 Juli 2021	11:31:49
38.	9.24	04 Juli 2021	11:31:50
39.	9.22	04 Juli 2021	11:31:51
40.	9.25	04 Juli 2021	11:31:52
41.	9.26	04 Juli 2021	11:31:54
42.	9.24	04 Juli 2021	11:31:55
43.	9.27	04 Juli 2021	11:31:56
44.	9.24	04 Juli 2021	11:31:58
45.	9.27	04 Juli 2021	11:31:59

Dari gambar yang telah dilampirkan, dapat disimpulkan bahwa pengujian nilai yang diatur pada Simulator dan dikirimkan ke *Ubidots* dapat terbaca dan berubah setiap 100 mili detik sekali, ada beberapa *sampling* data yang perubahannya atau *delaynya* selama 2 detik, error itu merupakan *error* yang disebabkan oleh jaringan wifi yang terkadang lemah sehingga mempengaruhi lamanya pengiriman data ke *Ubidots*

3.3 Pengujian Pada ESP32 Menggunakan Potensiometer



Gambar 9. Hasil pembacaan nilai pada ubidots dari ESP32 menggunakan potensiometer secara grafik

Pengujian Pada ESP32 menggunakan Potensiometer ditujukan untuk mengamati data dapat diterima atau tidak pada *Ubidots* serta dapat berubah sesuai pengaturan pada potensiometernya. Disini *sampling* data yang diperoleh yaitu setiap 100 milidetik.

Tabel 3. Data yang diterima ubidots dari ESP32 menggunakan potensiometer

No.	Nilai yang Terbaca	Tanggal Diterima	Waktu Diterima
1.	0.00	04 Juli 2021	10:48:05
2.	0.00	04 Juli 2021	10:48:05
3.	0.04	04 Juli 2021	10:48:06
4.	0.14	04 Juli 2021	10:48:07
5.	0.26	04 Juli 2021	10:48:08
6.	0.38	04 Juli 2021	10:48:09
7.	0.53	04 Juli 2021	10:48:09
8.	0.58	04 Juli 2021	10:48:10
9.	0.70	04 Juli 2021	10:48:11
10.	0.81	04 Juli 2021	10:48:12
11.	0.86	04 Juli 2021	10:48:13
12.	0.98	04 Juli 2021	10:48:14
13.	1.11	04 Juli 2021	10:48:15
14.	1.30	04 Juli 2021	10:48:15
15.	1.41	04 Juli 2021	10:48:16
16.	1.52	04 Juli 2021	10:48:17
17.	1.65	04 Juli 2021	10:48:18
18.	1.79	04 Juli 2021	10:48:19
19.	1.96	04 Juli 2021	10:48:19
20.	1.97	04 Juli 2021	10:48:20
21.	2.03	04 Juli 2021	10:48:21
22.	2.10	04 Juli 2021	10:48:22
23.	2.20	04 Juli 2021	10:48:23
24.	2.40	04 Juli 2021	10:48:24
25.	2.58	04 Juli 2021	10:48:24
26.	2.81	04 Juli 2021	10:48:25
27.	2.90	04 Juli 2021	10:48:26
28.	3.26	04 Juli 2021	10:48:27
29.	3.30	04 Juli 2021	10:48:28

Dari gambar yang telah dilampirkan, dapat disimpulkan bahwa pengujian nilai yang diatur pada Potensiometer dan dikirimkan ke *Ubidots* dapat terbaca dan berubah setiap 100 mili detik sekali, ada beberapa *sampling* data yang perubahannya atau *delaynya* selama 1 detik, error 1 detik itu merupakan *error yang* disebabkan oleh jaringan wifi yang terkadang lemah sehingga mempengaruhi lamanya pengiriman data ke *Ubidots*.

4. KESIMPULAN

Berdasarkan hasil pengujian dan analisis, maka dapat diperoleh kesimpulan:

1. PLC digunakan sebagai kontroler dan *Ubidots* sebagai platform IoT untuk menyimpan data, menampilkan grafik serta melakukan pengontrolan data dari simulator I/O dengan ESP32 sebagai perantara.
2. Komunikasi antara PLC dan ESP32 dapat dilakukan dengan membuat program pada Arduino IDE yang disesuaikan dengan Cx-Programmer. Serta pada hardware dihubungkan Rs485 pada PLC dan ESP32

3. Tegangan pada analog input dari simulator I/O dapat terbaca di *Ubidots*
4. Pada analog input, tegangan yang diukur 0-10V dengan *sampling* data sebesar 100 mili detik
5. Data dari analog input dapat terbaca pada *Ubidots* dengan *delay* kurang lebih 1 detik.

UCAPAN TERIMA KASIH

Saya Mahasiswa peneliti Tugas Akhir tahun akademik 2020/2021 mengucapkan terima kasih kepada Politeknik Negeri Bandung, melalui wakil Direktur Akademik atas bantuan pendanaan penyusunan tugas akhir nomor B/402/PL1.R1/EP.00.08/2021 kelompok A1.

DAFTAR PUSTAKA

- [1] Pangaribowo and H. Yulianda, "Sistem Monitoring Suhu Melalui Sistem Komunikasi Programable Logic Controller to Personal Computer," *Jurnal Teknologi Elektro*, vol. 7, no. 3, pp. 175-180, Sep. 2016.
- [2] K. Fayakun, A. Afandi, F. Afifah, and H. Ramza, "Universitas 17 Agustus 1945 Jakarta," p. 15, 2018.
- [3] F. Ardiansyah, M. F. Lawasi, and C. F. Hadi, "Sistem Monitoring Inkubator Penetas Telur Berbasis Android," vol. 01, p. 9, 2019.
- [4] M. C. Afrian, M. T. Asron and R. Wicaksono, "Prototipe Pengangkut Sampah Otomatis pada Pintu Ali dengan Sistem Informasi Menggunakan NodeMCU ESP8266 Berbasis PLC," *DOI*, vol. 5, no. 2, pp. 95-105, 2018.
- [5] S. Yahya, "Purwarupa SCADA PLTMH Menggunakan Komunikasi Modem GSM Berbasis PLC," *National Conference of Aplied Siences, Engineering, Business and Information Technology*, pp. 290-397, 2016.
- [6] P. Avina, H. Wicaksono and P. Santoso, "Sistem Keamanan Bangunan Multi Lokasi Berbasis IoT Menggunakan Siemens Logo! dan Raspberry Pi," *SinarFe7*, pp. 376-381, 2018.
- [7] Hermawan, Peggy Candra. "Perancangan Miniatur Mesin Pengisian Air Otomatis Menggunakan Arduino Nano Berbasis Internet of Things (IoT)." *Jurnal Online Mahasiswa (Jom) Bidang Teknik Elektro 1.1* (2020).
- [8] Wibowo, Dwiki Aditya, Suhanto Suhanto, and Darmadji Darmadji. "Rancang Bangun Implementasi Internet of Things Kontrol dan Monitoring Pada Generator Set 2 Kva." *Prosiding SNITP (Seminar Nasional Inovasi Teknologi Penerbangan)*. Vol. 3. No. 1. 2019.
- [9] Fernando, Rio, Ignatius Prasetya Dwi Wibawa, and Cahyanti Ekaputri. "Sistem Kendali dan Monitor Daya Listrik Pada Perangkat Listrik Rumah Berbasis IoT." *eProceedings of Engineering 5.3* (2018).
- [10] Shika, Sri Nur, Slamet Hariyadi, and Suhanto Suhanto. "Prototype Sistem Kontrol dan Monitoring Kubikel Menggunakan Konsep Internet of Thing Berbasis Mikrokontroler Arduino." *Prosiding Snitp (Seminar Nasional Inovasi Teknologi Penerbangan)*. Vol. 3. No. 1. 2019.