

Sistem Monitoring Kecepatan Motor dan Tekanan pada Saluran Air Berbasis Internet of Things (IoT)

Nasyith Hananur Rohiem¹, Novian Patria Uman Putra²

^{1,2}Teknik Elektro, Fakultas Teknik Elektro dan Teknologi Informasi, Institut Adhi Tama Surabaya

Email: ¹ nasyithhananur@itats.ac.id, ² novian111190@itats.ac.id

Abstract. *The problem that often occurs now is that the water pressure decreases in the waterways when several taps are opened so that it causes water filling which is quite long. For the pressure in the pipe to be constant, the solution to this problem is that a water pump system is needed to overcome the decreased water pressure when the number of taps opened varies which can be monitored remotely in the water installation system. In this final project, a system is designed to regulate the pressure of the water pump in the water pipe using three valves based on the Internet of Things (IoT). The pressure parameter is read using a pressure sensor whose output is processed and calibrated using the Arduino Uno. The output is used as a parameter to adjust the pump speed. To monitor this system using nodeMCU ESP8266 so that it can send data to android. Data from the monitoring sensor section and the controller will be processed by nodeMCU ESP8266, then the data will be published so that it can be monitored using Android. The display on this monitoring system is made on the MIT app inventor. Before turning on the system, the minimum, maximum, and sensor working limit setpoints are set. When the pressure is less than the setpoint, the motor speed will continue to increase. When the pressure is right at the set point the motor speed remains. When the pressure is more than the setpoint, the motor speed will decrease, and when the pressure exceeds the sensor working limit, the motor will stop. With this system, the pressure on the drinking water pipe can be stable as needed.*

Keywords: Water Pump, Monitoring, Internet of Things (IoT), Pressure Sensor, Stable Pressure

Abstrak. *Permasalahan yang sering terjadi sekarang ini adalah tekanan air yang menurun pada saluran air saat beberapa kran dibuka sehingga hal itu menyebabkan pengisian air yang cukup lama. Agar tekanan pada pipa konstan, maka solusi dari masalah ini yaitu diperlukan suatu sistem pompa air untuk mengatasi tekanan air yang menurun ketika jumlah keran yang dibuka bervariasi yang dapat dimonitoring dari jarak jauh pada sistem instalasi air. Pada Tugas Akhir ini dirancang sebuah sistem untuk mengatur tekanan pompa air pada saluran pipa air dengan menggunakan tiga valve berbasis Internet of Things (IoT). Parameter tekanan dibaca menggunakan sensor tekanan yang keluarannya diolah dan dikalibrasi menggunakan arduino uno. Hasil keluaran digunakan sebagai parameter untuk mengatur kecepatan pompa. Untuk memonitoring sistem ini menggunakan nodeMCU ESP8266 agar dapat mengirim data ke android. Data dari bagian sensor monitoring dan alat pengontrol akan diproses oleh nodeMCU ESP8266, kemudian data akan dipublikasikan sehingga dapat dipantau menggunakan android. Tampilan pada sistem monitoring ini dibuat pada mit app inventor. Sebelum menyalakan sistem maka diatur set point tekanan minimal, maksimal, dan batas kerja sensor. Ketika tekanan kurang dari set point yang ditentukan, maka kecepatan motor akan terus bertambah. Ketika tekanan tepat pada set point maka kecepatan motor tetap. Ketika tekanan lebih dari set point maka kecepatan motor akan berkurang, dan ketika tekanan melebihi batas kerja sensor maka motor akan mati. Dengan sistem ini maka tekanan pada pipa air minum bisa stabil sesuai kebutuhan.*

Kata Kunci: Pompa Air, Monitoring, Internet of Things (IoT), Sensor Pressure. Tekanan Stabil

1. Pendahuluan

Air merupakan kebutuhan pokok yang harus dipenuhi dan sangat dibutuhkan setiap hari oleh masyarakat Indonesia. Kebutuhan air masyarakat Indonesia sekitar 125–150 liter per hari setiap orangnya, yang jika dikalkulasikan lagi sekitar 625–1500 liter per hari untuk tingkat konsumsi keluarga (Suparno, Ono & Suprihatin, Suprihatin. (2013)). Karena keterbatasan suplai dan kebutuhan air yang cukup banyak, mengakibatkan tekanan suplai air dari pusat menjadi sangat kecil. Tekanan air akan menurun pada saluran air saat beberapa kran dibuka. Resikonya ketika pompa air bekerja pada kecepatan maksimum namun tidak ada

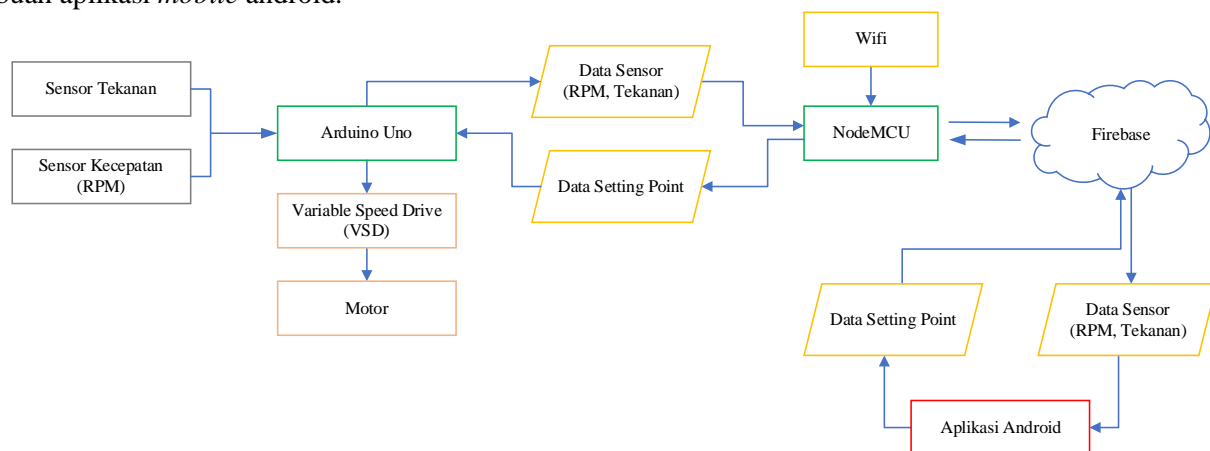
air yang mengalir menuju konsumen, maka tekanan pada pipa akan semakin besar, ketika tekanan di pipa terus naik hal ini akan dapat merusak pipa dan juga pompa air itu sendiri. Begitu juga ketika kebutuhan air akan turun namun realitanya kebutuhan sedang naik, maka aliran air yang mengalir pada konsumen akan kecil akibat dari tekanan pada pipa yang kecil. Hal ini mengakibatkan distribusi air menjadi tidak beraturan (Wibowo, I. (2017))

Dari beberapa permasalahan di atas dan beberapa referensi dari penelitian sebelumnya (Setiadi, D., & Muhaemin, M. N. (2018); Sasmoko, Dani. (2020); Tanuatmadja, Regan & Wijono, F.X.. (2018); Wibowo, I. (2017)), maka diperlukan suatu sistem pompa air untuk mengatasi tekanan air yang menurun ketika jumlah keran yang dibuka bervariasi yang dapat dimonitoring dari jarak jauh pada sistem instalasi air. Pada penelitian ini dirancang sebuah sistem untuk mengatur tekanan pompa air pada saluran pipa air berbasis *Internet of Things* (IoT). Parameter tekanan dibaca menggunakan sensor tekanan yang keluarannya diolah dan dikalibrasi menggunakan arduino. Hasil keluaran digunakan sebagai parameter untuk mengatur kecepatan pompa sehingga diharapkan tekanan dalam pipa air minum bisa stabil sesuai kebutuhan. Untuk memonitoring sistem ini menggunakan nodeMCU ESP8266 agar dapat mengirim data ke *firebase* agar dapat di akses oleh aplikasi android.

Sistem monitoring (Triyono, B., et al. (2021)) kecepatan pompa dan tekanan pada saluran pipa air berbasis IoT ini diharapkan dapat membantu mempermudah pemantauan tekanan pada saluran pipa air yang dapat dimonitoring dari jarak jauh menggunakan android dan nantinya sistem ini bisa menjadi solusi untuk memperbaiki distribusi air agar lebih efisien.

2. Metodologi

Pada penelitian disusun atas beberapa perangkat menjadi sebuah blok diagram seperti yang ditunjukkan pada gambar 1. Pada gambar 1 terdapat 2 sensor sebagai *input*, 2 perangkat kontroler dengan fungsi yang berbeda. Terdapat 1 *output* pada gambar 1 tersebut yaitu untuk kendali aktuator dan 1 jaringan komunikasi 2 arah yang terhubung dengan *firebase* agar dapat diakses secara *online* dan ditampilkan dalam sebuah aplikasi *mobile android*.

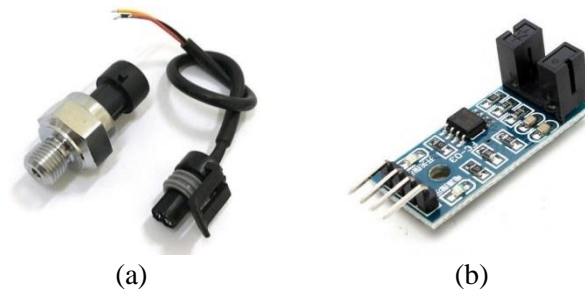


Gambar 1. Blok diagram sistem

2.1. Sensor

Terdapat 2 jenis sensor yang digunakan untuk menunjang penelitian ini, yaitu sensor *pressure transmitter* dan sensor kecepatan motor seperti yang ditunjukkan pada gambar 2(a) dan 2(b). Sensor *pressure transmitter* seperti yang ditunjukkan pada gambar 2 adalah sensor tekanan yang dilengkapi rangkaian pengkondisi sinyal, sehingga sinyal dari sensor tekanan dapat ditransmisikan ke komputer. Luaran dari sensor ini adalah sinyal listrik DC, yang dapat berupa tegangan atau arus listrik, untuk kemudian ditransmisikan melalui penghantar dengan jangkauan jarak sesuai dengan kualitas penghantar tersebut dan kuantitas sinyal yang ditransmisikan. *Pressure Transmitter* dikemas di dalam satu komponen yang terbuat dari *stainless steel*. Sedangkan konektornya terdiri dari 3 pin, 2 buah untuk *input* dan 1 untuk *output*. *Housing* dari *Pressure Transmitter* ini harus tertutup rapat agar didapatkan hasil stabil, linear dan nilai histeresis dapat diabaikan serta tahan tekanan dan guncangan. Sedangkan sensor kecepatan pada penelitian ini menggunakan optocoupler

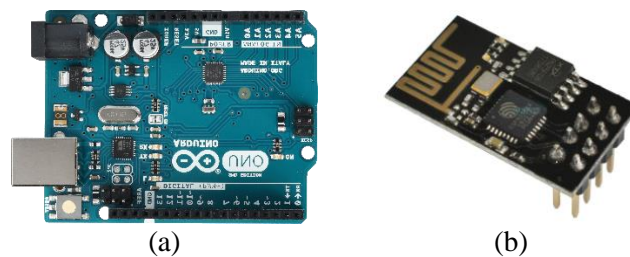
dengan rotary encoder yang menghasilkan output sinyal 0 dan 1 yang selanjutnya akan dilakukan *counter* pembacaan kondisi dalam satuan waktu tertentu yang kemudian di konversi dalam bentuk kecepatan rotasi.



Gambar 2 (a) Sensor *Pressure Transmitter*, (b) Sensor kecepatan motor

2.1. Kontroler

Terdapat 2 kontroler yang digunakan pada penelitian ini yaitu Arduino Uno dan NodeMCU. Kedua kontroler ini merupakan platform yang bersifat *opensource* dan di program menggunakan aplikasi *software* yang sama. Arduino uno dan NodeMCU seperti pada gambar 3(a) dan 3(b) memiliki fitur dan kapabilitas yang hampir sama yaitu sebagai kontroler. Namun NodeMCU memiliki keunggulan yang mampu mengakses jaringan internet nirkabel sehingga mampu terhubung dengan fitur tambahan seperti database ataupun platform data yang mengharuskan terhubung jaringan internet yang lain.



Gambar 3 (a) Arduino uno, (b) NodeMCU ESP8266

Pada penelitian ini kedua kontroler tersebut memiliki fungsi yang berbeda-beda. Arduino uno memiliki fungsi untuk membaca data sensor kemudian mengirimkan data tersebut ke NodeMCU, serta melakukan pengaturan kecepatan motor dengan kondisi berdasarkan pembacaan sensor serta menerima data dari NodeMCU berupa *setting point* kondisi pembacaan sensor tekanan. Sedangkan NodeMCU pada penelitian ini digunakan untuk mengirimkan data pembacaan sensor dari arduino menuju database agar data tersebut dapat diakses oleh aplikasi mobile sehingga data yang dihasilkan dapat di akses kapan saja melalui jaringan internet serta memberikan *update setting point* sensor tekanan yang berasal dari aplikasi android dan dikirim kepada arduino.

2.3. Variable Speed Drive (VSD)

VSD seperti yang ditunjukkan pada gambar 4 merupakan sebuah alat pengatur kecepatan motor AC dengan mengubah nilai frekuensi dan tegangan yang masuk ke motor. pengaturan nilai frekuensi dan tegangan ini dimaksudkan untuk mendapatkan kecepatan putaran dan torsi motor yang di inginkan atau sesuai dengan kebutuhan. Secara sederhana prinsip dasar inverter untuk dapat mengubah frekuensi menjadi lebih kecil atau lebih besar yaitu dengan mengubah tegangan AC menjadi tegangan DC kemudian dijadikan tegangan AC lagi dengan frekuensi yang berbeda atau dapat diatur.



Gambar 4. Variable Speed Drive

Pada penelitian ini VSD difungsikan untuk mengatur kecepatan motor pompa agar tekanan angin di dalam pipa dapat dikondisikan ketika tekanan air yang akan dialirkan terlalu rendah. Pengaturan kecepatan motor pada VSD dilakukan dengan mengirimkan sinyal level tegangan pada VSD yang kemudian pada sistem VSD akan di konfersi kedalam level kecepatan motor. Sinyal level tegangan yang menjadi input VSD pada penelitian ini dihasilkan oleh sinyal *pulse width modulation* (PWM) dari kontroler arduino kemudian dikonversi menjadi sinyal level tegangan.

2.4. Firebase

Firebase dengan logo seperti yang ditunjukkan pada gambar 5 adalah penyedia layanan *realtime* database. Suatu aplikasi yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada cloud Firebase. Firebase memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, Ios, Javascript, Java, Objective-C dan Node JS. Firebase *client library* yang sudah diterapkan pada aplikasi yang dibangun yang akan mengambil data secara *realtime*.



Gambar 5. logo Firebase

2.5. MIT App Inventor

MIT App Inventor merupakan platform untuk memudahkan proses pembuatan aplikasi android sederhana tanpa harus mempelajari atau menggunakan bahasa pemrograman yang terlalu banyak. App ini dapat mendesain aplikasi android sesuai keinginan dengan menggunakan berbagai macam *layout* dan komponen yang tersedia. Pada MIT App Inventor terdapat dua halaman utama, yaitu halaman *designer* dan halaman *blocks*. Halaman desain digunakan untuk mendesain tampilan aplikasi dengan berbagai komponen dan layout yang disediakan sesuai dengan keinginan. Sedangkan halaman *blocks* digunakan untuk memprogram jalannya aplikasi android sesuai dengan tujuan.

3. Hasil dan Analisa

Dari proses yang sudah dilakukan diatas terdapat beberapa data hasil yang dilakukan pengujian dan analisa. Pertama dilakukan terlebih dahulu pengujian pada bagian sensor. Sensor tekanan diuji dengan membuka dan tutup 3 *valve* yang telah terpasang pada saluran pipa hasil pengujian seperti yang ditunjukkan pada tabel 1. Untuk sensor kecepatan diuji dengan melakukan variasi kecepatan pada motor dengan hasil pembacaan yang ditunjukkan pada tabel 2. Kedua sensor tersebut juga telah dibandingkan dengan alat ukur yang sesuai pada 5 kali percobaan dan mampu menghasilkan rata-rata *error* yang kecil yaitu 1,4% untuk sensor tekanan dan 0,54% untuk sensor kecepatan motor.

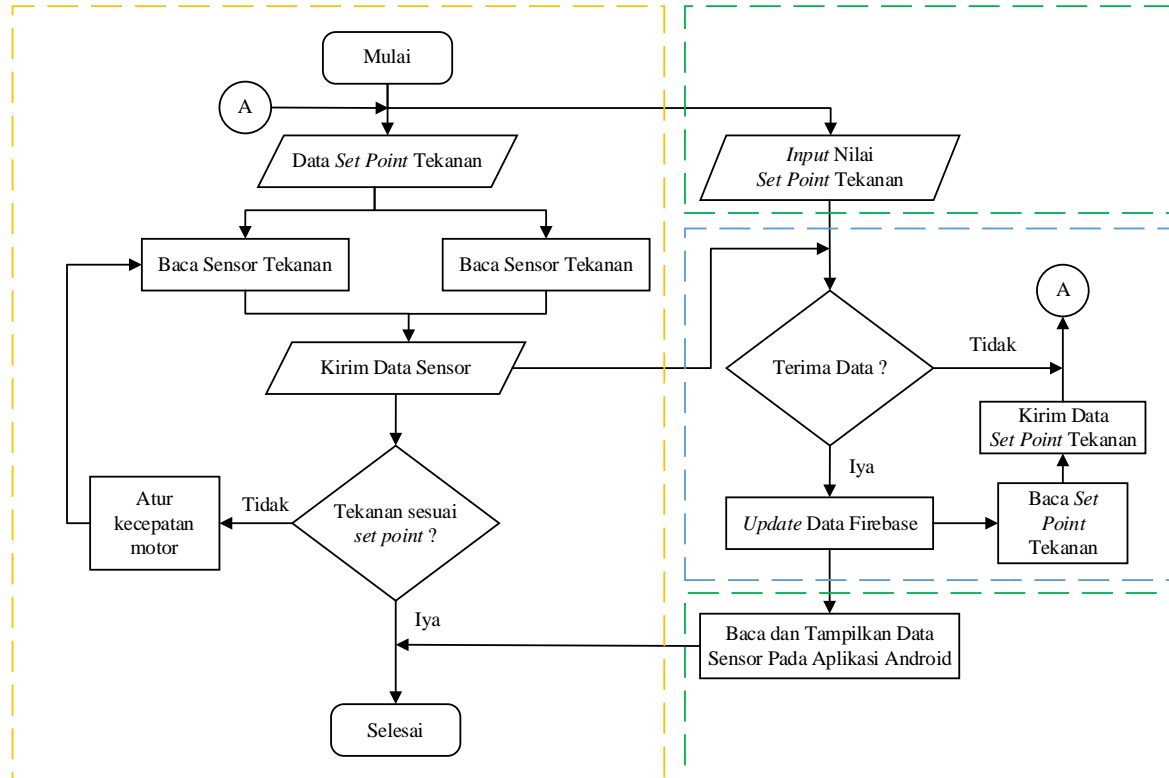
Tabel 1 Pengujian sensor tekanan

No	Kondisi Pengujian	Hasil pembacaan (psi)		Error (%)
		Sensor tekanan	Alat ukur	
1	Valve 1,2,3 tertutup	27,94	27,50	1,60
2	Valve 1 terbuka	19,38	19	2,00
3	Valve 1,2 terbuka	18,19	18,4	1,14
4	Valve 1,3 terbuka	18,25	18,4	0,82
5	Valve 1,2,3 terbuka	17,24	17	1,41

Tabel 2 Pengujian sensor kecepatan motor

No	Level tegangan input VSD (Volt)	Hasil pembacaan (rpm)		Error (%)
		Sensor optocoupler	Alat ukur	
1	3,3	1874	1881	0,37
2	4	2275	2280	0,22
3	2,75	1565	1568	0,19
4	3,4	1936	1938	0,10
5	1,5	848	855	0,83

Pada bagian kontroler pengujian dilakukan berdasarkan kinerja yang harus diproses, proses yang terjadi di dalam kontroler itu sendiri disusun berdasarkan algorithma yang dikembangkan berdasarkan diagram alir yang ditunjukkan pada gambar 6. Pada kontroler arduino uno mampu melakukan fungsi pembacaan sensor dengan baik, fungsi pengiriman data pembacaan sensor ke NodeMCU, menerima data dari NodeMCU serta arduino juga mampu melakukan kontrol kecepatan dengan mengirim sinyal ke VSD hingga tekanan pada pipa mampu mencapai *set point* yang diinginkan. Untuk waktu yang dibutuhkan agar tekanan sesuai dengan *set point* seperti yang ditunjukkan pada tabel 3. Hasil dari tabel 3 menunjukkan sinyal kontrol kecepatan yang dikirimkan ke VSD dari arduino merespon dengan baik dengan kondisi semua *valve* terbuka sehingga waktu *setling point* yang dicapai cukup cepat antara 2-3 detik sehingga tekanan dapat terjaga dengan baik.


Gambar 6 Diagram alir sistem (blok kuning merupakan proses didalam arduino, hijau merupakan proses didalam aplikasi android, dan biru merupakan proses didalam NodeMCU)
Tabel 3 Pengujian *setling point*

No	Set point tekanan (psi)	RPM motor	Setling time (s)
1	18	954	3
2	17,5	863	2
3	20	1015	3
4	19	976	2
5	17	844	2

Pada NodeMCU juga mampu melakukan proses upload dan proses baca data pada database dengan baik serta mengirimkan data *set point* tekanan pipa ke arduino sesuai dengan yang telah di *setting* pada aplikasi android. Pada tabel 4 menunjukkan proses kesesuaian penerimaan dan pengiriman data yang dilakukan antara aplikasi android, Firebase, NodeMCU, dan Arduino. Kesesuaian data yang diproses menunjukkan hasil yang bagus karena tidak terdapat *error* data atau 0%.

Tabel 4 pengujian komunikasi data

No	Kondisi Pengujian		App Android	Firebase	NodeMCU	Arduino	Error (%)
1	Pengiriman data <i>set point</i> (psi)	Nilai <i>set point</i> tekanan udara	18	18	18	18	0
2			19	19	19	19	0
3			17,5	17,5	17,5	17,5	0
4			18,5	18,5	18,5	18,5	0
5			20	20	20	20	0
No	Kondisi Pengujian		Arduino	NodeMCU	Firebase	App Android	Error (%)
1	Pengiriman data pembacaan sensor	Nilai sensor tekanan (psi)	18	18	18	18	0
		Nilai sensor kecepatan motor (rpm)	954	954	954	954	0
2		Nilai sensor tekanan (psi)	17,5	17,5	17,5	17,5	0
		Nilai sensor kecepatan motor (rpm)	863	863	863	863	0
3		Nilai sensor tekanan (psi)	20	20	20	20	0
		Nilai sensor kecepatan motor (rpm)	1015	1015	1015	1015	0
4		Nilai sensor tekanan (psi)	19	19	19	19	0
		Nilai sensor kecepatan motor (rpm)	976	976	976	976	0
5		Nilai sensor tekanan (psi)	17	17	17	17	0
		Nilai sensor kecepatan motor (rpm)	844	844	844	844	0

4. Kesimpulan

Dari hasil penelitian yang telah dilakukan pengujian dan juga analisa pada hasil pengujiannya maka dapat diambil kesimpulan bahwa

1. Sensor tekanan dan kecepatan motor mampu bekerja dengan baik dengan *error* rata-rata 1,3% dan 0,54% setelah dibandingkan dengan alat ukur pada 5 kali pengambilan data.
2. *Setling time* yang dibutuhkan sistem untuk mencapai *set point* tekanan yang diinginkan cukup cepat berkisar 2 sampai 3 detik pada 5 kali percobaan pergantian *set point* tekanan.
3. Komunikasi data berjalan dengan baik, proses pengiriman data hingga penerimaan data dari Aplikasi android menuju arduino dan sebaliknya menghasilkan *error* 0% karena tidak ada data tidak sesuai.
4. Jaringan internet berpengaruh terhadap proses komunikasi data terutama pada waktu proses *update* data *firebase*.

Referensi

- Suparno, Ono & Suprihatin, Suprihatin. (2013). Teknologi Proses Pengolahan Air untuk Mahasiswa dan Praktisi Industri. Rakhmawati, R. (2015). Prototipe Pengaturan Tekanan Air pada Sistem Distribusi Air. *MediaTeknika Jurnal Teknologi*, 72-82. BIBLIOGRAPHY \I 1057
- Setiadi, D., & Muhaemin, M. N. (2018). Penerapan Internet of Things (IoT) pada Sistem Monitoring Irigasi (Smart Irigasi).
- Sasmoko, Dani. (2020). Sistem Monitoring aliran air dan Penyiraman Otomatis Pada Rumah Kaca Berbasis IoT dengan Esp8266 dan Blynk. *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*. 4. 1. 10.22373/crc.v4i1.6128.
- Tanuattmadja, Regan & Wijono, F.X.. (2018). Perancangan Sistem Monitoring dan Controlling Pompa Air secara Wireless Berbasis Android. *TESLA: Jurnal Teknik Elektro*. 19. 124. 10.24912/tesla.v19i2.2695.
- Triyono, B., et al. (2021). "Electrical Motor Interference Monitoring Based On Current Characteristics." *Journal of Physics: Conference Series* 1845(1): 012044.
- Wibowo, I. (2017). Sistem Pemantau Ketinggian Air Nirkabel. *TELEKONTRAN*, 49-53.