

SIMULASI *CONDITION MONITORING* MESIN BERBASIS *IOT* MENGUNAKAN ESP32

¹Al Kautsar Permana, ²Adam Satria, dan ³Hikari Qurrata'ain Nurhadi, ⁴Archie Farras
Belmino Tyto Putra

^{1,2,3,4}Program Studi Teknologi Rekayasa Otomotif, Politeknik STMI Jakarta, Indonesia

¹alkautsarpermana@kemenperin.go.id, ²adam.satria@kemenperin.go.id,

³hikariqurata@kemenperin.go.id, ⁴putra.tyto12@gmail.com

ABSTRACT

This research develops an IoT-based machine condition monitoring system capable of real-time monitoring of machine power consumption to enhance operational efficiency and predictive maintenance in industrial settings. The system utilizes the ACS712 current sensor to detect electrical current consumed by the machine. Using the Thingspeak platform, data collected from the sensor is periodically sent to a cloud server and displayed in a dashboard format, enabling remote monitoring. The monitoring program is designed to reduce the data load on the server, with data being transmitted every 10 seconds to minimize irrelevant data redundancy. In testing, the system successfully transmitted data to the Thingspeak server stably, despite occasional connectivity disruptions between the microcontroller and server. Over a total testing duration of one hour, 100 data samples were successfully uploaded, showing an average current of 2.8 Amperes and power of 550 Watts, with an operational voltage range of 220-235 Volts.

Keywords: ESP32, Condition Monitoring, IoT connectivity, Embedded System, Thingspeak

ABSTRAK

Seiring dengan meningkatnya penggunaan mesin CNC di industri, kebutuhan untuk melakukan *condition monitoring* pada level mesin juga semakin tinggi. Tuntutan *condition monitoring* secara *real-time* menjadi kebutuhan dalam efisiensi biaya, terutama perawatan *predictive* dan *preventive*. Penelitian ini mengembangkan sistem *monitoring* kondisi mesin berbasis *IoT* yang mampu memantau konsumsi daya mesin secara *real-time* guna meningkatkan efisiensi operasional dan pemeliharaan prediktif dalam industri. Sistem ini memanfaatkan sensor arus ACS712 untuk mendeteksi arus listrik yang dikonsumsi oleh mesin. Melalui *platform Thingspeak*, data yang diperoleh dari sensor dikirim secara berkala ke *server cloud* dan ditampilkan dalam bentuk *dashboard* yang memudahkan pemantauan jarak jauh. Program *monitoring* ini dirancang untuk mengurangi beban data pada *server* di mana data dikirim setiap 10 detik untuk meminimalkan redundansi data yang tidak relevan. Dalam pengujian, sistem berhasil mengirimkan data ke *server Thingspeak* dengan stabil, meskipun terdapat gangguan konektivitas sesekali antara *microcontroller* dan *server*. Dari total waktu pengujian selama satu jam, sebanyak 100 sampel data berhasil diunggah, menunjukkan rata-rata arus yang dihasilkan sebesar 2,8 Ampere dan daya sebesar 550 Watt, dengan tegangan operasional dalam rentang 220-235 Volt.

Kata Kunci: ESP32, Condition Monitoring, Konektivitas *IoT*, Embedded System, Thingspeak

LATAR BELAKANG

Mesin *Computer Numerical Control (CNC)* merupakan peralatan manufaktur yang sangat umum digunakan di perusahaan-perusahaan manufaktur modern. Keberadaan mesin ini telah merevolusi proses produksi dengan kemampuan untuk mengolah data secara akurat dan cepat, sehingga proses pemesinan benda kerja dapat dilaksanakan dengan presisi tinggi. CNC memanfaatkan kode numerik (NC) yang dibuat menggunakan perangkat lunak *computer aided manufacturing (CAM)*, yang kemudian dapat langsung ditransfer ke mesin CNC untuk menjalankan operasi pemesinan secara otomatis. Dengan kemampuan ini, mesin CNC tidak hanya meningkatkan efisiensi produksi tetapi juga meningkatkan kualitas produk yang dihasilkan. Akurasi tinggi dan kecepatan proses yang ditawarkan oleh mesin CNC menjadikannya pilihan utama dalam berbagai aplikasi manufaktur, mulai dari industri

otomotif, penerbangan, hingga peralatan medis. Namun, seiring dengan meningkatnya penggunaan mesin CNC di industri, kebutuhan untuk melakukan *condition monitoring* pada level mesin juga semakin tinggi. *Condition monitoring* adalah proses pengawasan kondisi operasional mesin untuk mendeteksi potensi kerusakan atau keausan sebelum menyebabkan kegagalan yang signifikan. Dalam konteks mesin CNC, *monitoring* aktivitas dan performa mesin menjadi area riset yang sangat penting.

Menurut penelitian yang dilakukan juga oleh *Cleginaldo Pereira de Carvalho* (2020), dimana Arduino digunakan untuk mengontrol sensor *ultrasonic*, sensor warna, dan *accelerometer* untuk memonitor *coolant* mesin dan vibrasi *gearbox* [5]. Penelitian dalam bidang ini bertujuan untuk mengembangkan sistem *monitoring* yang dapat memberikan informasi *real-time* mengenai kondisi mesin, sehingga tindakan perawatan *preventif* dapat dilakukan lebih dini. Hal ini tidak hanya membantu dalam mencegah *downtime* yang tidak terencana, tetapi juga meningkatkan efisiensi operasional dan umur pakai mesin CNC. Hal ini tentunya berbeda dengan sistem *monitoring* konvensional yang umumnya menggunakan check sheet yang dicatat manual oleh operator pada rentang waktu tertentu sesuai dengan jadwal inspeksi. Kendati demikian, riset terkait *monitoring* mesin CNC untuk melihat okupansi mesin CNC secara harian belum dilakukan dan hal ini merupakan celah riset yang diangkat pada studi ini. Ruang lingkup riset ini mencakup berbagai aspek teknis, termasuk pengembangan sistem *internet of things (IoT)* [3] untuk memonitor status mesin CNC ke dalam *dashboard IoT* yang dapat diakses dari manapun. Integrasi mesin CNC dengan sistem *IoT* sudah pernah dilakukan oleh [4]. *microcontroller Phyton* digunakan untuk *monitoring* vibrasi dan akselerasi mesin CNC dan membandingkannya dengan acuan standar ISO 10816.

Pada riset ini, kami menerapkan purwarupa sistem IoT untuk memonitor kondisi pemakaian mesin CNC berbasis simulasi Wokwi secara *real-time*. Data pemakaian mesin CNC dapat membantu manajemen dalam melihat efisiensi penggunaan mesin secara baik. Perencanaan penggunaan mesin dan pengelolaannya dapat dimaksimalkan untuk keuntungan perusahaan. Data pemakaian dapat diambil dari lampu indikator status mesin CNC yang dikoneksikan dengan *platform IoT* sehingga status apakah mesin sedang digunakan atau tidak dapat monitor. Purwarupa yang dibuat harus memiliki *low-complexity* dan *low-cost* agar purwarupa sistem *condition monitoring* dapat diimplementasikan bagi perusahaan manufaktur. Oleh karena itu, perangkat lunak sistem *microcontroller* dipilih menggunakan *platform open-source* dan perangkat kerasnya menggunakan *microcontroller* yang sederhana seperti ESP32 yang memiliki kapabilitas untuk terkoneksi ke jaringan internet.

METODE PENELITIAN

Proses pembuatan purwarupa sistem *condition monitoring* mesin CNC diberikan pada gambar 1. Pembuatan purwarupa dibagi menjadi 4 pekerjaan utama. Pertama adalah menentukan *microcontroller* dan merancang sistem kontrol agar sensor dapat membaca arus pada mesin. Proses kedua adalah membuat program *microcontroller* agar tersambung dengan *platform IoT*. Sistem komunikasi *web service* menggunakan *application programming interface (API)* yang tersedia pada *platform IoT* seperti Thingspeak dipakai sebagai jembatan antara perangkat keras dengan *server IoT*. Ketiga, membuat *wiring*

diagram dalam aplikasi Wokwi untuk disimulasikan dengan parameter yang sudah dirancang. Proses terakhir adalah pengujian dengan cara membaca data real-time dari sistem kontrol dan mengirimkannya ke *server IoT* untuk melihat konsumsi arus dan daya dalam rentang waktu tertentu. Data yang dikumpulkan oleh sensor tersebut kemudian diproses oleh *microcontroller* ESP32. ESP32 dapat dikonfigurasi untuk melakukan analisis dasar seperti pemfilteran data dan penghitungan statistik sederhana (*mean, peak, RMS*). Data yang lebih kompleks dapat dikirim ke komputer atau *server* untuk analisis lebih lanjut menggunakan perangkat lunak khusus.



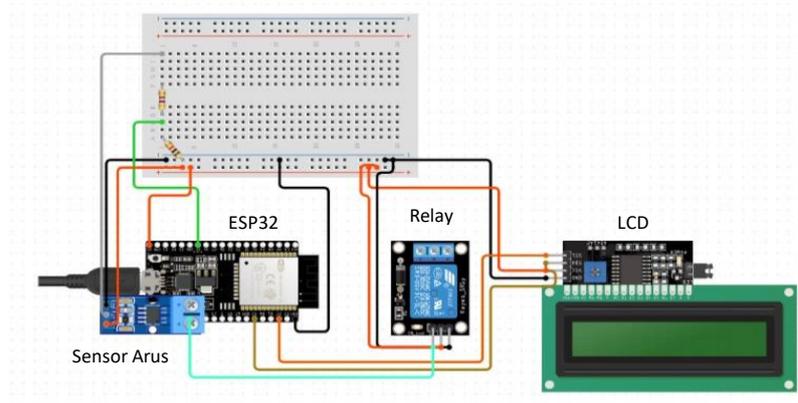
Gambar 1. Diagram blok metodologi penelitian

1. Perancangan sistem kontrol

Perangkat keras dalam penelitian ini terdiri dari *microcontroller* ESP32, *input*, dan *output*. *Microcontroller* yang digunakan akan memroses *input*, yaitu sensor arus, lalu hasil dari *output* akan ditampilkan pada LCD serta ke *server Thingspeak* berupa arus dan daya listrik. Sensor arus terhubung ke relay 5V yang memiliki peran untuk menghubungkan perangkat ke listrik alternating current (AC) dalam hal ini lamp indicator mesin CNC. Status sensor dan hasil pembacaan sensor ditampilkan pada perangkat LCD untuk memudahkan proses *debugging* sistem. Sensor arus memiliki tegangan operasional pada 5V sedangkan *microcontroller* beroperasi pada 3.3V. *microcontroller* bisa saja mendapat tegangan 5V apabila dihubungkan ke *power supply* eksternal, namun dalam penelitian ini tegangan operasional mengikuti bawaan *microcontroller* yaitu 3.3.V. Agar sensor arus dapat terhubung, maka dibuatlah rangkaian pembagi tegangan dengan menggunakan 2 buah resistor. Kedua resistor ini dirangkai pada *breadboard* dengan hubungan seri. Secara umum, luaran dari rangkaian pembagi tegangan dituliskan sebagai berikut:

$$V_{out} = \frac{R_2}{(R_1 + R_2)} \times V_{in}$$

dimana V_{out} merupakan output 3.3V dan V_{in} adalah tegangan operasional sensor arus 5V. Resistor R1 dan R2 dipilih sedemikian agar menghasilkan V_{out} 3.3V. Sistem *monitoring* akan dibuat berbasis simulasi dengan bantuan Wokwi, dimana seluruh proses wiring serta program dibuat melalui Wokwi. *Platform* ini dapat terkoneksi dengan berbagai *server IoT* seperti *Thingspeak* atau *Nodered*. *Wiring diagram* sistem *monitoring* ditunjukkan pada gambar 2. Detail dari setiap *wiring diagram* diberikan pada tabel 1. Sensor arus membaca arus dari mesin melalui relay. Pin *load* dari sensor arus terhubung dengan pin IN dari relay, sedangkan DC+ dan DC- relay terhubung ke VIN dan GND ESP32. Kabel pada mesin terhubung ke pin relay NO dan COM.



Gambar 2. Wiring diagram sistem IoT

Tabel 1. Konfigurasi pin pada ESP32 dengan LCD dan sensor arus

Pin ESP 32	Pin LCD	Pin Sensor Arus
VIN	VCC	VCC
GND	GND	GND
D21	SDA	
D22	SCL	
D34		OUT

2. Pembuatan program dan fungsi

Ketika mesin CNC beroperasi, sensor arus akan membaca arus kabel catu daya mesin. Sensor arus terkoneksi dengan ESP32 melalui relay sebagai sumber daya dan input data digital. data tersebut diteruskan dan diunggah ke *cloud server Thingspeak*. Ketika data sudah disimpan dalam *database server*. Dalam program ini, proses pengembangan diawali dengan deklarasi library, yang mencakup pengaturan komunikasi serial, pengendalian layar LCD, serta deklarasi variabel lokal yang diperlukan untuk menyimpan data sementara. Selain itu, API key dan koneksi Wi-Fi juga diatur untuk memastikan komunikasi yang lancar antara *microcontroller* dan *server Thingspeak*.

Setelah tahap inisialisasi selesai, *microcontroller* akan membaca data dari sensor arus yang terhubung, dimana hasil pengukuran tersebut kemudian ditampilkan pada LCD sebagai indikator langsung. Selanjutnya, data yang telah dibaca dikirimkan secara real-time ke *server Thingspeak* untuk diproses lebih lanjut atau disimpan dalam *database cloud*. Proses ini tidak hanya memungkinkan *monitoring* secara lokal melalui LCD, tetapi juga secara global melalui internet, memanfaatkan *platform Thingspeak* untuk analisis data lebih lanjut dan pemantauan jarak jauh.

HASIL DAN PEMBAHASAN

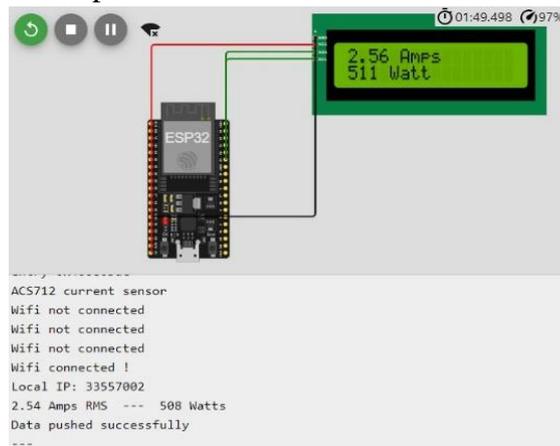
Sistem yang sudah dibuat memiliki beberapa konfigurasi disisi perangkat lunak dan perangkat keras. Di sisi perangkat lunak, kalibrasi sensor perlu dilakukan terlebih dahulu agar sensor menampilkan hasil pembacaan yang tepat. Nilai minimum 4096 harus dideklarasikan sebagai nilai minimum pembacaan sensor karena resolusi *analog-to-digital conversion* dari *microcontroller* yang digunakan. Tabel 2 menunjukkan algoritma untuk kalibrasi sensor arus. Sensor arus disesuaikan dengan tegangan operasional *microcontroller* yaitu 3.3V.

Tabel 2. Algoritma kalibrasi dan fungsi tegangan *peak-to-peak*

Algoritma 1: Kalibrasi dan fungsi tegangan *peak-to-peak*

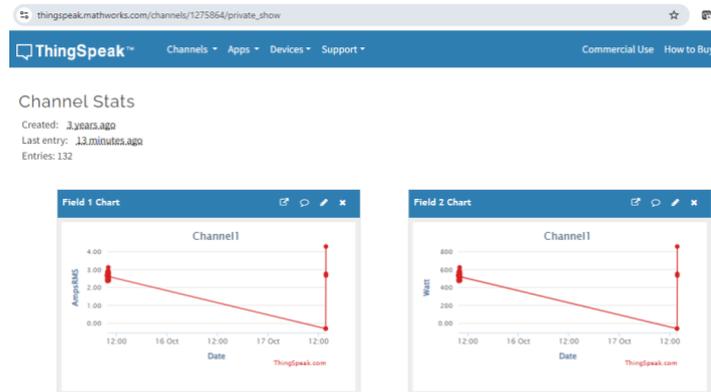
Input: V
Data: maxValue, minValue
// maxValue: data yang terbaca dari sensor
//minValue: nilai 4096 resolusi ADC pada ESP32
if V > maxValue
| V == maxValue;
if V < minValue
| V == minValue;
end
result ← hasil kalibrasi tegangan *peak-to-peak*
result = ((maxValue – minValue) x 3.3) x 4096

Sistem diuji dengan mengunggah program komputer ke dalam *microcontroller*. Selanjutnya *microcontroller* akan mengaktifkan koneksi ke jaringan *Wi-Fi* dengan credentials *Wi-Fi* yang sudah diisi pada program. Setelah berhasil koneksi dengan *Wi-Fi*, *microcontroller* akan mencoba koneksi *API* dengan *server thingspeak*. Apabila berhasil terkoneksi, maka data akan dilempar dari *microcontroller* ke *server*. Sebaliknya, kegagalan koneksi akan mengakibatkan data tidak berhasil terkirim ke *server*. Kendati demikian, LCD tetap akan menampilkan hasil pembacaan sensor.



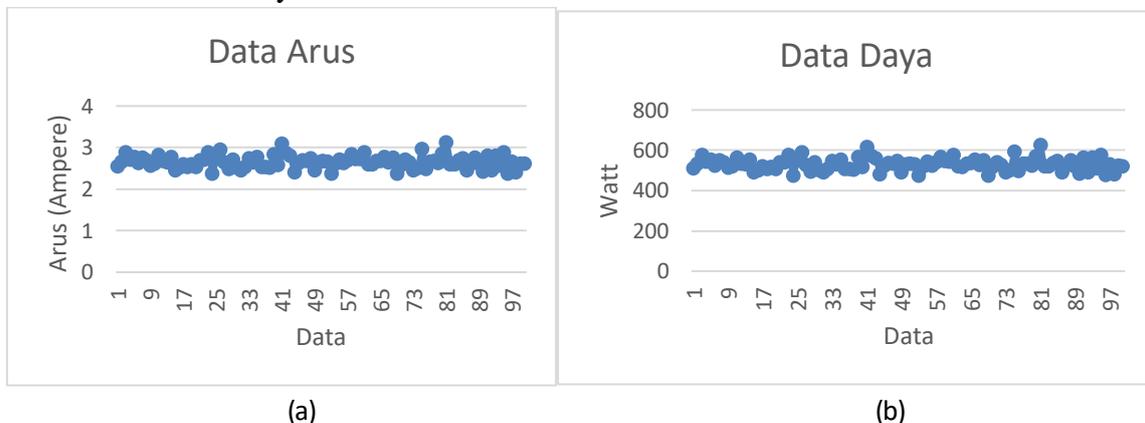
Gambar 3. Pengujian sistem monitoring IoT

Tipikal sistem *IoT* adalah membaca sensor dan melemparkan datanya ke *server* setiap detik. Kondisi ini akan membuat pemrosesan data di *server* akan membesar dan mayoritas data merupakan data yang tidak memiliki arti untuk sistem keputusan bisnis. Oleh karena itu, dalam pengujian sistem ini data dikirim tiap 10 detik ke *server IoT* untuk mengurangi beban di *server*. Pertimbangan didasarkan agar dalam satu menit setidaknya ada 6 data yang merekam kondisi riil mesin dan mencatatnya ke dalam *server*, sehingga sedikit anomali yang terjadi pada mesin dapat terekam tanpa banyak redundansi data. Halaman *channel IoT* sistem ditunjukkan pada gambar 4. Grafik sebelah kiri menunjukkan data arus sedangkan grafik sebelah kanan adalah daya yang dikonsumsi mesin. *Server IoT* akan merekam data secara terus menerus. Apabila tidak ada data yang dikirimkan pada selang waktu tertentu, maka data akan dianggap nol dan data tersebut terus terekam. Ketika ada data baru muncul, maka data otomatis muncul sebagai entri baru. Berdasarkan gambar 4, dapat dilihat bahwa terdapat data 15 Oktober, kemudian data kosong pada 16 Oktober. Entri data baru muncul kembali ditanggal 17 Oktober.



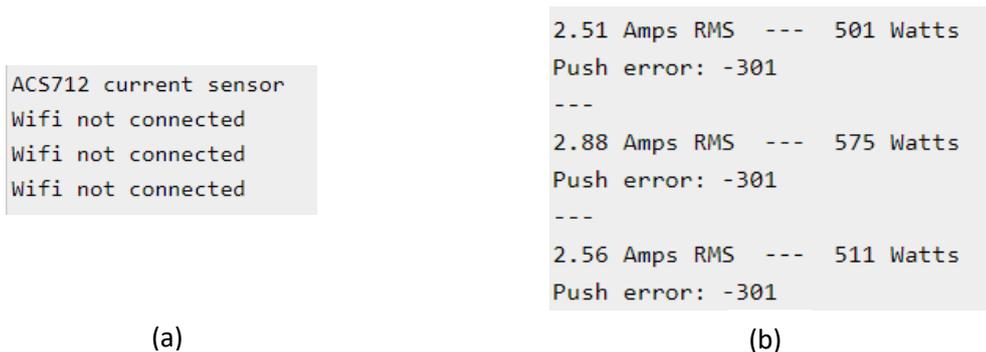
Gambar 4. Dashboard thingspeak yang menunjukkan data arus pada field 1 dan daya pada field 2

Data yang sudah masuk ke *server IoT* akan selamanya tersedia pada *server* tersebut. Data tersebut dapat diexport ke dalam file berformat .csv untuk diolah lebih lanjut atau dapat divisualisasikan menggunakan matlab yang tersedia pada *server*. Sensor membaca data arus yang kemudian dihitung pula dayanya. Kedua data ini yang diupload ke *server IoT*. Saat pengujian sistem, *microcontroller* diaktifkan hingga mengirimkan data sebanyak 100 sampel selama kurang lebih 1 jam, setelah itu sistem dimatikan. Idealnya, data yang harus terbaca dalam satu jam adalah 360 sampel. Namun, Wokwi tidak akan mengirimkan data jika pada saat simulasi halaman Wokwi tidak tampil. Kondisi aktual pada saat simulasi, halaman Thingspeak, halaman wokwi, dan wiring diagram diobservasi secara bergantian sehingga pada saat Wokwi tidak ditampilkan, data tidak terkirim ke *server IoT*. Agar probabilitas data yang terkirim tinggi, selama simulasi halaman Wokwi perlu dibuka secara terus menerus. Selain masalah diatas, terdapat masalah teknis lainnya yang dijelaskan pada bagian lain dalam penelitian ini. Gambar 5a dan 5b menunjukkan data arus dan daya secara berturut-turut. Secara umum, data arus memiliki rata-rata sebesar 2.8 Ampere dan daya rata-rata sebesar 550 Watt dengan tegangan operasional direntang 220-235 Volt. Dari grafik, bisa disimpulkan bahwa mesin dalam keadaan operasional secara kondisi normal. Apabila pada grafik arus mengalami lonjakan/*Spike* diatas rata-rata, ada kemungkinan besar bahwa motor listrik pada mesin CNC mengalami kondisi abnormal. Jika arus listrik mesin mengalami penurunan namun konstan, menunjukkan bahwa mesin sedang tidak beroperasi, namun mesin dibiarkan menyala.



Gambar 5. Data yang dikirim ke *server IoT* (a) data arus, dan (b) data daya

Konektivitas sistem tidak seutuhnya handal. Dari pengujian sistem yang dijalankan selama kurang lebih 1 jam. Hanya terdapat 100 sampel data yang berhasil dikirim ke *server*. Selama dijalankan, ada waktu sekitar 10-15 menit koneksi internet terputus antara *microcontroller* dengan Wi-Fi. Pada perangkat LCD terus menampilkan data yang dibaca sensor setiap 10 detik namun data tersebut tidak masuk ke *server IoT*. Selain itu, masalah koneksi terjadi pada API yang tidak saling terjalin meskipun koneksi internet terhubung. Terkait masalah koneksi API, setidaknya ada 8 data yang gagal terkirim ke *server IoT*. Gambar 6 menunjukkan dua jenis masalah koneksi yang terjadi pada sistem saat pengujian.



Gambar 6. Data log dari serial monitor (a). koneksi Wi-Fi tidak terhubung dengan *microcontroller* dan (b) API tidak mengirimkan data ke *server IoT*

KESIMPULAN DAN SARAN

Penelitian ini berhasil membuat simulasi sistem *condition monitoring* berbasis *IoT* untuk mesin CNC menggunakan ESP32 dan simulator Wokwi. Integrasi perangkat keras dengan *platform IoT* dapat dilakukan melalui simulator Wokwi. Dalam perancangan sistem, pengujian secara *real-time* sebelumnya secara konvensional menggunakan metode *trial and error* dengan menggunakan perangkat keras, sekarang dapat dilakukan tanpa harus memiliki perangkat keras seperti ESP32 beserta sensor dan LCDnya. Fasilitas ini dapat menjadi sarana pengembangan purwarupa sistem *monitoring* berbasis *IoT* sebelum sarana dan prasarana sistem dibeli. Simulasi seperti ini akan meningkatkan keberhasilan implementasi perangkat fisik karena secara program dan wiring dapat divalidasi terlebih dahulu menggunakan Wokwi.

REFERENSI

- [1] Aragón González, G., Barragán Santiago, I., Bautista Godínez, I., Becerra Martínez, P., Delgado Román, O., León Galicia, A., & Manrique Garay, J. (2022). Remote control and monitoring of a hydraulic machine. In *Journal of Physics: Conference Series* (Vol. 2307). Institute of Physics. <https://doi.org/10.1088/1742-6596/2307/1/012009>.
- [2] Anandh, R., & Indirani, G. (2018). Real Time Health Monitoring System Using Arduino with Cloud Technology. *Asian Journal of Computer Science and Technology*, 7(S1), 29–32. <https://doi.org/10.51983/ajst-2018.7.s1.1810>
- [3] Mabrouki, J., Azrou, M., Dhiba, D., Farhaoui, Y., & Hajjaji, S. E. (2021). IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. *Big Data Mining and Analytics*, 4(1), 25–32. <https://doi.org/10.26599/BDMA.2020.9020018>.

- [4] Al-Naggar, Y. M., Jamil, N., Hassan, M. F., & Yusoff, A. R. (2021). Condition monitoring based on IoT for predictive maintenance of CNC machines. In *Procedia CIRP* (Vol. 102, pp. 314–318). Elsevier B.V. <https://doi.org/10.1016/j.procir.2021.09.054>
- [5] Carvalho, C. P. de, Simões, B. dos R., Carvalho, É. A. de, & Diniz, R. R. (2020). SENSING: An Approach to Establish the First Step to Prepare a CNC Machine for IoT Implementation. *International Journal of Advanced Engineering Research and Science*, 7(11), 71–79. <https://doi.org/10.22161/ijaers.711.10>
- [6] Abu Sneineh, A., & Shabaneh, A. A. A. (2023). Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things. *MethodsX*, 11. <https://doi.org/10.1016/j.mex.2023.102401>
- [7] Budijono, S., & Felita. (2021). Smart Temperature Monitoring System Using ESP32 and DS18B20. In *IOP Conference Series: Earth and Environmental Science* (Vol. 794). IOP Publishing Ltd. <https://doi.org/10.1088/1755-1315/794/1/012125>
- [8] Pravalika, V., & Rajendra Prasad, C. (2019). Internet of things-based home monitoring and device control using Esp32. *International Journal of Recent Technology and Engineering*, 8(1 Special Issue 4), 58–62.
- [9] Julián, R. C. L., Oscar, D. F. C., & Jaramillo-Matta, A. A. (2023). PULSE OXIMETER IMPLEMENTED WITH ESP32 AND MONITORED IN BLYNK AND THINKSPEAK. *ARNP Journal of Engineering and Applied Sciences*, 18(14), 1692–16999. <https://doi.org/10.59018/0723210>
- [10] Sarker, S., Rakib, M. A., Islam, S., & Shafin, S. S. (2021). An IoT-based Smart Grid Technology: Bidirectional Power Flow, Smart Energy Metering, and Home Automation. In *2021 International Conference on Maintenance and Intelligent Asset Management, ICMIAM 2021*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ICMIAM54662.2021.9715188>