

# Rancang Bangun *Prototype* Pengenal Sensor Semi Otomatis pada Perangkat *Internet Of Things*

Dzulfikar<sup>1</sup>, Nur Rohman Rosyid<sup>1,\*</sup>, Ronald Adrian<sup>1</sup>

<sup>1</sup>Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;  
dzulfikar.sv@mail.ugm.ac.id

ronald.adr@ugm.ac.id

\*Korespondensi: nrohmanr@ugm.ac.id;

**Abstract** – *The industrial revolution 4.0 had a huge impact. The industrial revolution 4.0 combines machines or devices, computers and the internet into one unified system. One of the technologies that has developed due to the industrial revolution is the Internet of Things (IoT). Currently, IoT technology has been widely implemented in various sectors, starting from the mining sector, supply chain management, agriculture, transportation, and so on. Nowadays, the implementation of IoT is still done manually. This application can cause problems such as requiring more time and increasing human error. Therefore, we need a system to overcome this. This research was conducted to automatically detect and recognize the type of sensor that is installed or added. The research carried out was implemented using the Micropython framework. The tests carried out focus on system functionality. Based on the research conducted, the system successfully detects and recognizes sensors semi-automatically. The detection and configuration results are stored in the microcontroller in the form of a json file and the results are displayed via a web app.*

**Keywords** – *IoT, Sensor, Semi-automatic, Recognition, Micropython*

**Intisari** – Revolusi industri 4.0 memberi dampak yang sangat besar. Revolusi industri 4.0 menggabungkan mesin atau perangkat, komputer, dan internet menjadi satu sistem yang utuh. Salah satu teknologi yang berkembang karena adanya revolusi industri yaitu *Internet of Things (IoT)*. Saat ini, teknologi IoT sudah banyak diimplementasikan di berbagai sektor, mulai dari sektor pertambangan, *supply chain management*, pertanian, transportasi, dan lain sebagainya. Pada masa sekarang, penerapan penentuan sensor pada perangkat IoT masih dilakukan secara manual dengan menuliskannya secara langsung pada kode program. Penerapan ini dapat menimbulkan permasalahan seperti membutuhkan waktu lebih dan meningkatnya *human error*. Oleh karena itu, dibutuhkan suatu sistem untuk mengatasi hal tersebut. Penelitian ini dilakukan untuk mendeteksi dan rekognisi secara otomatis jenis sensor yang dipasang atau ditambahkan. Penelitian yang dilakukan diimplementasikan dengan menggunakan *framework Micropython*. Pengujian yang dilakukan berfokus pada fungsionalitas sistem. Berdasarkan penelitian yang dilakukan, sistem berhasil mendeteksi dan mengenali sensor secara semi otomatis. Pendeteksian dilakukan dengan pengguna menekan tombol *scan* pada tampilan *web app*. Hasil deteksi dan konfigurasi disimpan di dalam *microcontroller* dalam bentuk file json dan menampilkan data hasil deteksi melalui *web app*.

**Kata kunci** – *IoT, Sensor, Semi-otomatis, Pengenalan, Micropython*

## I. PENDAHULUAN

Revolusi industri 4.0 memberi dampak yang sangat besar. Revolusi industri 4.0 menggabungkan mesin atau perangkat, komputer, dan internet menjadi satu sistem yang utuh. Salah satu teknologi yang berkembang karena adanya revolusi industri yaitu *Internet of Things (IoT)* [1].

IoT merupakan suatu sistem yang terdiri sekumpulan perangkat yang dapat berkomunikasi, bertukar informasi dan dapat diakses melalui internet [2]. Saat ini, teknologi IoT sudah banyak diimplementasikan di berbagai sektor, mulai dari sektor pertambangan, *supply chain management*, pertanian, transportasi, dan lain sebagainya [3].

Pada masa sekarang, penerapan IoT masih dilakukan secara manual. Sebagai contoh untuk mengimplementasikan teknologi IoT di bidang pertanian, untuk melakukan monitoring ataupun mengontrol suatu perangkat di lahan pertanian perlu berhubungan secara langsung dalam memprogram perangkat *microcontroller*, menentukan pin sensor yang akan dipakai, dan mengunggah program yang dibuat melalui kabel USB. Hal ini tidak akan menjadi masalah apabila sistem IoT yang dibuat digunakan selamanya dan dipastikan tidak ada perubahan. Namun, apabila akan dilakukan pembaharuan terhadap kode program atau

menambah sensor yang akan digunakan, perlu mengulang dari awal. Selain hal ini membutuhkan banyak waktu, pembaharuan secara manual ini juga dapat meningkatkan adanya *human error*. Oleh sebab itu, penelitian ini dilakukan sebagai solusi untuk mengatasi hal tersebut.

Luaran dari penelitian ini berupa *prototype* perangkat IoT yang dapat melakukan deteksi dan rekognisi secara semi otomatis jenis sensor yang dipasang atau ditambahkan. Dengan adanya *prototype* ini diharapkan dapat memberikan kemudahan dalam implementasi teknologi IoT, memudahkan dalam melakukan manajemen perangkat, serta memudahkan dalam melakukan pembaharuan program secara terpusat melalui jaringan.

## II. DASAR TEORI

### A. SD-IoT

SD-IoT atau *Software Defined Network – Internet of Thing* merupakan kombinasi penerapan SDN pada sistem IoT. Penerapan SDN pada sistem IoT diketahui sangat aplikatif dalam implementasi sistem IoT skala besar [4]. SDN menyediakan metode yang efisien dalam melakukan kontrol perangkat IoT secara terpusat, manajemen jaringan, dan mendukung layanan yang tersedia [5] [6].

B. *Wireless Sensor Network*

*Wireless Sensor Network* merupakan kumpulan dari sensor independen terdistribusi yang terhubung pada jaringan nirkabel dengan tujuan melakukan pemantauan fisik atau kondisi lingkungan. Pemantauan yang dilakukan dapat berupa suhu lingkungan, kelembaban udara, tingkat polusi, dan lain sebagainya. WSN ini terdiri dari set sensor yang dapat saling berkomunikasi, dan bertukar informasi serta data [7].

C. Python

Python merupakan bahasa pemrograman tingkat tinggi, *interpreter*, dan berorientasi objek yang dapat digunakan pada berbagai sektor, seperti dalam pembuatan web, *back-end server*, kecerdasan buatan, data sains, dan lain sebagainya [8]. Karakteristik kemudahan dalam penggunaan, *library* yang *powerful*, dan kemudahan dalam membaca kode menjadikan bahasa Python salah satu bahasa pemrograman yang paling banyak diimplementasikan [9].

D. Micropython

Micropython merupakan implementasi bahasa Python yang dioptimalisasi agar dapat berjalan pada *microcontroller* dan perangkat *embedded system* [10]. *Firmware* Micropython menyediakan module yang berfungsi untuk mengakses fungsionalitas perangkat keras seperti GPIO, I2C, *Timers*, ADC, DAC, PWM, SPI, CAN, Bluetooth, WiFi dan USB. Micropython ini dapat diinstal pada perangkat *microcontroller* seperti ESP32, ESP8266, STM32, SAMD, Raspberry Pi, dan lain sebagainya [11].

E. Microdot

Microdot merupakan *framework web server* berbasis bahasa Python yang dirancang agar dapat berjalan pada perangkat dengan sumber daya terbatas seperti *microcontroller*. *Framework* Microdot ini terinspirasi dari *framework web server* Flask. Microdot berjalan sesuai dengan standard Bahasa pemrograman Python dan Micropython [12].

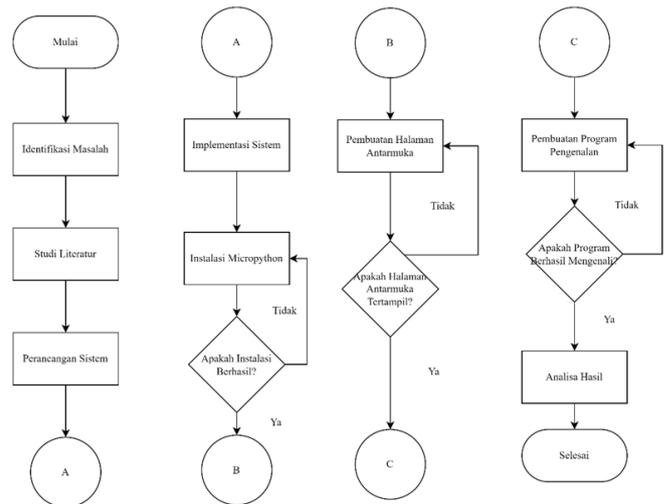
F. ESP32

ESP32 merupakan *microcontroller System on Chip (SoC)* rendah daya, terjangkau, yang dilengkapi dengan konektivitas jaringan seperti WiFi dan Bluetooth serta perangkat tambahan lainnya [13]. *Microcontroller* ini dikeluarkan oleh Espressif Systems dengan jumlah inti sebanyak dua buah dengan arsitektur Harvard dan dapat diaplikasikan pada aplikasi IoT, *wearable electronic*, perangkat *mobile*, dan lain sebagainya [14].

III. METODOLOGI

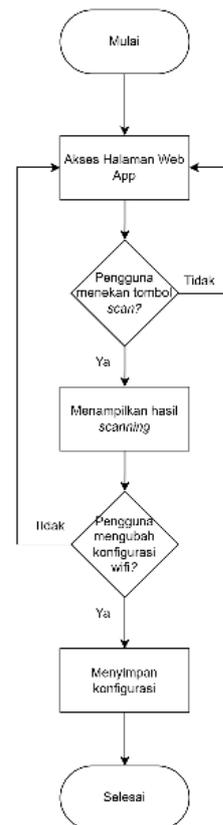
A. Tahapan Penelitian

Tahapan penelitian yang dilakukan peneliti dibagi menjadi enam tahap. Tahapan penelitian ini digunakan sebagai acuan dalam melakukan penelitian. Setiap tahap yang dilakukan memiliki fungsi dan tujuan masing-masing. Visualisasi dari setiap tahapan penelitian yang dilakukan dapat dilihat pada Gambar 1.

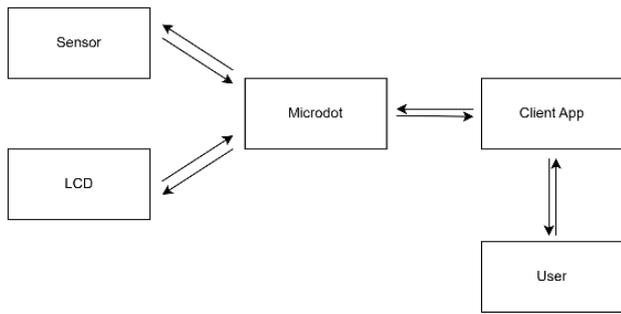


Gambar 1. Diagram alir tahapan penelitian

Diagram alir sistem dan *firmware* yang digunakan dapat dilihat pada Gambar 2 dan Gambar 3. Seperti pada Gambar 2, pendeteksian dilakukan setelah peneliti menekan tombol *scan*. Hasil dari pemindaian yang dilakukan selanjutnya akan ditampilkan melalui *web app*.



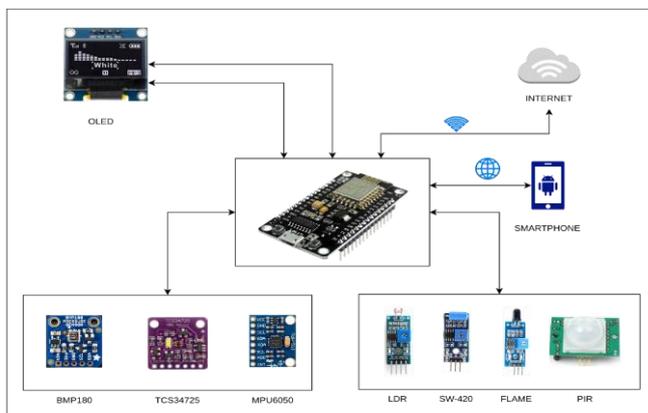
Gambar 2. Diagram alir sistem



Gambar 3. Diagram alir *firmware*

B. Gambaran Umum Rancangan Sistem

Rancangan sistem dibuat setelah mendapatkan informasi yang dibutuhkan pada langkah sebelumnya. Gambaran umum dari rancangan sistem pada penelitian ini seperti pada Gambar 4. Sistem yang dibuat terdiri dari komponen *microcontroller*, layar oled, sensor i2c, sensor gpio, dan perangkat *smartphone*.



Gambar 4. Gambaran umum rancangan sistem

C. Tahapan Pengujian

Tahap pengujian sistem dilakukan untuk memastikan perangkat yang telah dibuat dapat bekerja sebagaimana mestinya. Pengujian dilakukan dengan metode *grey box testing* dan berfokus pada fungsionalitas perangkat. Pengujian *grey box testing* merupakan pengujian yang dilakukan dengan sedikit informasi terkait dengan perangkat. Hasil dari pengujian sistem dicatat dan didokumentasikan untuk selanjutnya dilakukan analisa.

IV. HASIL DAN PEMBAHASAN

A. Tampilan Fisik Perangkat

Tampilan fisik dari perangkat yang dibuat dapat dilihat pada Gambar 5. Perangkat ini terdiri dari beberapa bagian sebagai berikut.

- Layar Oled berfungsi untuk menampilkan informasi alamat IP dari perangkat.
- Tombol Reset digunakan untuk mereset ulang perangkat apabila terdapat *error* dan memuat ulang perangkat.

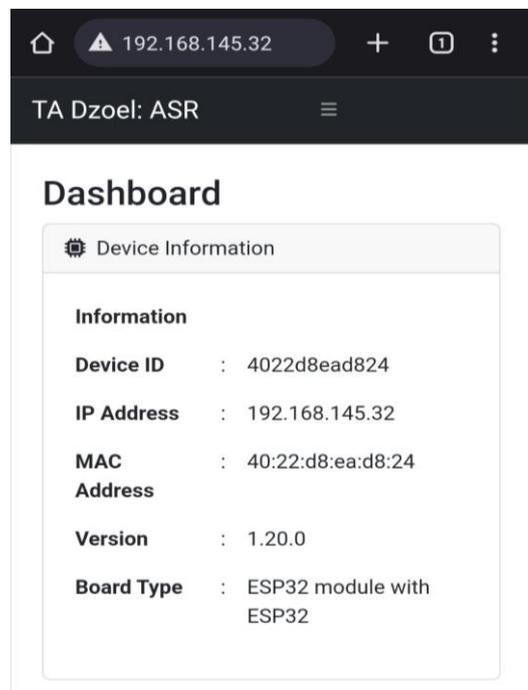
- Konektor sensor I2C digunakan sebagai tempat untuk memasang sensor berjenis I2C. Konektor ini terdiri dari 4 pin yaitu GND, VCC, SCL, dan SDA.
- Konektor sensor GPIO digunakan sebagai tempat untuk memasang sensor yang bersifat umum atau *general*. Konektor ini terdiri dari 3 pin yaitu GND, VCC, dan DATA (Analog / Digital).
- Konektor *Power Supply* digunakan untuk menyuplai daya perangkat. Tegangan yang dapat diterima oleh perangkat dengan menggunakan konektor ini berkisar 6 – 16 Volt.
- Konektor *Type C* digunakan untuk menyuplai daya perangkat menggunakan kabel *type C*.
- Konektor *Micro USB* digunakan untuk menyuplai daya perangkat menggunakan kabel *micro USB*.



Gambar 5. Tampilan fisik perangkat

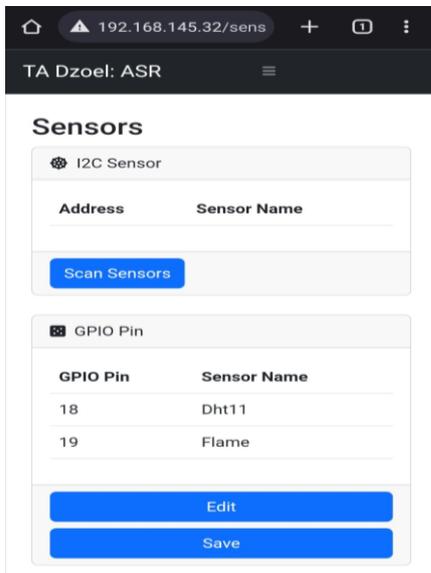
B. Tampilan Antarmuka Perangkat

Tampilan halaman utama seperti terlihat pada Gambar 6. Pada halaman utama ini terdapat menu utama yang berada di sebelah kiri dan tampilan informasi detail perangkat. Halaman utama memuat informasi terkait ID *chip*, alamat IP dari perangkat, *mac address*, versi Micropython, dan tipe *board*.



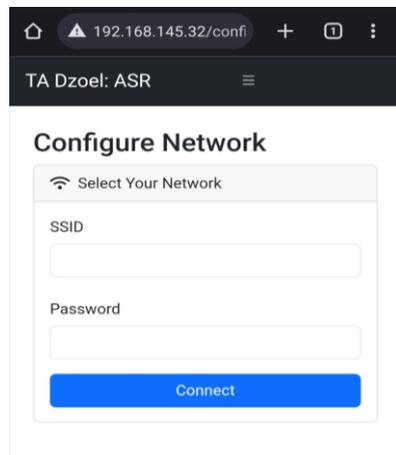
Gambar 6. Tampilan halaman utama

Halaman sensor menampilkan detail informasi sensor yang terpasang atau ditambahkan. Informasi yang ditampilkan berupa detail nama dan alamat *hex* dari sensor I2C yang terpasang, serta nama dan pin dari GPIO sensor. Tampilan halaman sensor terlihat seperti pada Gambar 7.



Gambar 7. Tampilan halaman sensor

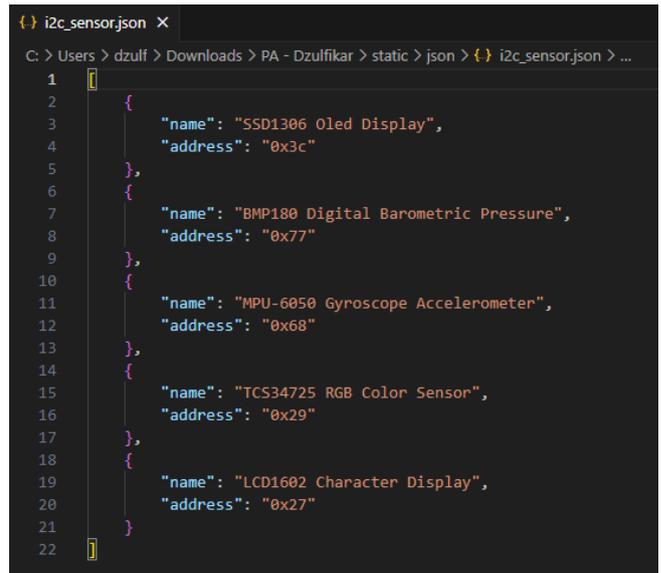
Halaman konfigurasi seperti terlihat pada Gambar 8 digunakan untuk mengatur konektivitas perangkat. Konfigurasi yang dapat dilakukan pada halaman ini yaitu konfigurasi SSID dan Password agar perangkat dapat terhubung ke jaringan internet.



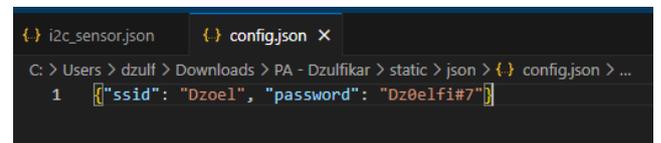
Gambar 8. Tampilan halaman konfigurasi

C. Hasil Pengujian

Pengujian yang dilakukan pada penelitian ini berfokus pada pengujian fungsionalitas perangkat. Pengujian ini dilakukan untuk memastikan perangkat dapat bekerja dan berfungsi dengan baik tanpa ada kendala. Pengujian yang dilakukan pada penelitian ini tertampil pada Tabel 1. Sensor yang berhasil dideteksi dan konfigurasi *wifi* pada pengujian ini disimpan dalam file json seperti pada Gambar 9 dan Gambar 10.



Gambar 9. Hasil pemindaian sensor



Gambar 10. Hasil konfigurasi *wifi*

Tabel 1. Hasil pengujian

Nama Pengujian	Bagian Yang Diuji	Langkah Pengujian	Hasil Pengujian
Nyala	Nyala perangkat	Menghubungkan perangkat dengan sumber listrik menggunakan power supply, kabel type C, dan kabel Micro USB	Berhasil
Reset	Tombol Reset	Menekan tombol reset	Berhasil
Oled	Layar Oled	Menghidupkan dan mematikan perangkat	Berhasil
Access Point	AP Mode	Menghubungkan <i>smartphone</i> pada SSID perangkat	Berhasil
Antarmuka	Halaman antarmuka	Mengakses halaman antarmuka menggunakan alamat IP	Berhasil
Scan I2C	Tombol scan	Menekan tombol <i>scan</i>	Berhasil
GPIO sensor	Gpio sensor	Menambahkan sensor pada tabel	Berhasil

## V. SIMPULAN

Sistem pendeteksi sensor semi otomatis dibuat dengan memanfaatkan teknologi IoT. Sistem pendeteksi ini memanfaatkan *firmware* Micropython dan *framework* Microdot yang secara khusus dibuat untuk dapat digunakan pada perangkat *low level*. Berdasarkan pengujian yang telah dilakukan, perangkat yang dibuat dapat diakses pengguna melalui web browser pada *smartphone*. Data konfigurasi *wifi* dan *general* sensor dapat disimpan oleh *microcontroller* dalam bentuk file berformat json.

## REFERENSI

- [1] E. A. Alaref and S. A. Khan, "Industry 4.0 and its Technologies: A Systematic Literature Review," *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 1004-1009, 2021.
- [2] A. Mathur, A. Dabas and N. Sharma, "Evolution From Industry 1.0 to Industry 5.0," *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pp. 1390-1394, 2022.
- [3] M. Aazam, S. Zeadally and K. A. Harras, "Deploying Fog Computing in Industrial Internet of Things and Industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674 - 4682, 2018.
- [4] S. Chilibingua, S. Manzano, P. Córdova and M. V. Garcia, "An Approach of Low-cost Software-Defined Network (SDN) Based Internet of Things," *2020 International Conference of Digital Transformation and Innovation Technology (Incodtrin)*, pp. 70-74, 2020.
- [5] S. A. Arman, M. M. Rahman, S. F. Rahman, N. P. Urmi, P. P. Urmee, N. Muslim and S. Islam, "Developing an IoT Networks-based Testbed for Software-Defined Networks," *2020 IEEE Region 10 Symposium (TENSymp)*, pp. 1752-1755, 2020.
- [6] T. M. Chau Nguyen, D. B. Hoang and T. D. Dang, "A software-defined model for IoT clusters: Enabling applications on demand," *2018 International Conference on Information Networking (ICOIN)*, pp. 776-781, 2018.
- [7] M. S. Obaidat and S. Misra, "Introduction to Wireless Sensor Networks." *Principles of Wireless Sensor Networks*, Cambridge: Cambridge University Press, 2014.
- [8] A. Kumar and S. P. Panda, "A Survey: How Python Pitches in IT-World," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 248-251, 2019.
- [9] A. Nagpal and G. Gabrani, "Python for Data Analytics, Scientific and Technical Applications," *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 140-145, 2019.
- [10] G. Gaspar, P. Fabo, M. Kuba, J. Flochova, J. Dudak and Z. Florkova, "Development of IoT applications based on the MicroPython platform for Industry 4.0 implementation," *2020 19th International Conference on Mechatronics - Mechatronika (ME)*, pp. 1-7, 2020.
- [11] Micropython, "MicroPython," 2023. [Online]. Available: <http://docs.micropython.org/en/latest/#>. [Accessed 2023].
- [12] M. Grinberg, "Microdot," 2021. [Online]. [Accessed 25 July 2023].
- [13] M. Babiuch, P. Foltýnek and P. Smutný, "Using the ESP32 Microcontroller for Data Processing," *2019 20th International Carpathian Control Conference (ICCC)*, pp. 1-6, 2019.
- [14] H. Kareem and D. Dunaev, "The Working Principles of ESP32 and Analytical Comparison of using Low-Cost Microcontroller Modules in Embedded Systems Design," *2021 4th International Conference on Circuits, Systems and Simulation (ICCSS)*, pp. 130-135, 2021.