



# Rancang Bangun *System Monitoring Purgung Pada Bag Filter Berbasis IoT (Internet of Things)*

Alfi Sihab<sup>1,2</sup>, Sonki Prasetya<sup>3\*</sup>, Rum Karimak<sup>2</sup>

<sup>1</sup> Program Studi Teknik Mesin - EVE, Jurusan Teknik Mesin, Politeknik Negeri Jakarta, Kampus UI Depok, 16424

<sup>2</sup> Electric & Instrument Raw Mill & Kiln Tuban 2 Department, PT Solusi Bangun Indonesia Pabrik Tuban

<sup>3</sup> Program Studi Teknologi Rekayasa Konversi Energi, Jurusan Teknik Mesin, Politeknik Negeri Jakarta, Kampus UI Depok, 16424

---

## Abstrak

*Kegiatan monitoring purging pada bag filter merupakan salah satu kegiatan pemantauan proses produksi yang dilakukan oleh tim patrol hampir di setiap pergantian shift. Saat ini kegiatan monitoring dilakukan secara manual oleh patroller, yaitu hanya dengan mengandalkan indra pendengaran untuk mengetahui unit purging aktif atau tidak. Cara ini memerlukan waktu yang lama dan kemungkinan terjadi kesalahan juga tinggi yang membuat waktu monitoring juga semakin panjang. Untuk mengatasi masalah ini, muncul gagasan untuk membuat alat monitoring purging berbasis Internet of Things yang dilakukan secara realtime. Proses yang dilakukan untuk merealisasikan sistem monitoring ini diantaranya, melakukan analisis pemilihan sensor dan kontroler. Dilanjut dengan pembuatan PCB, pembuatan program, tampilan antarmuka pada web Thinger.IO. Waktu pengiriman data dari kontroler ke web diatur 0.5 detik sekali. Total data yang tersimpan sebesar 96.6% dari total data yang terkirim. Dari hasil pemantauan selama alat berjalan, rata-rata suhunya adalah 43.76°C.*

*Kata kunci : NodeMCU ESP8266, bag filter, purging, thinger.io, interrupt*

## Abstract

*Purgung monitoring activities on bag filters is one of the production process monitoring activities carried out by the patrol team at almost every shift change. Currently monitoring activities are carried out manually by the patroller, that is, only by relying on the sense of hearing to determine whether the purging unit is active or not. This method takes a long time and the possibility of errors is also high which makes the monitoring time also longer. To overcome this problem, the idea emerged to create a purging monitoring tool based on the Internet of Things that is carried out in real time. The processes carried out to realize this monitoring system include analyzing the selection of sensors and controllers. Followed by PCB creation, program creation, interface display on the Thinger.IO web. The data transmission time from the controller to the web is set once every 0.5 seconds. The total data stored is 96.6% of the total data sent. From the results of monitoring while the tool is running, the average temperature is 43.76°C.*

*Keywords : NodeMCU ESP8266, bag filter, purging, thinger.io, interrupt*

---

\* Corresponding author *E-mail address*: [sonki.prasetya@mesin.pnj.ac.id](mailto:sonki.prasetya@mesin.pnj.ac.id)

## 1. PENDAHULUAN

*Bag filter* merupakan alat yang berfungsi untuk memisahkan partikel kering dari gas (udara) pembawanya. pada bagian dalam *bag filter*, sirkulasi gas kotor dan partikel masuk ke dalam beberapa longsongan filter (kantong atau *cloth bag*) yang terpasang secara paralel, kemudian meninggalkan debu pada filter tersebut[1]. Partikel debu tertahan di sisi kotor kain, sedangkan gas bersih akan melewati sisi bersih kain. Debu secara periodik dibersihkan dari kantong menggunakan guncangan atau memakai sirkulasi udara terbalik tergantung pada tipenya, maka bisa dikatakan bahwa *bag filter* adalah alat yang menerima gas yang mengandung debu, menyaringnya, mengumpulkan debunya, kemudian mengeluarkan gas yang bersih ke atmosfer.

Penggunaan *IoT (Internet of Things)* pada industri masih sangat jarang. Salah satu penelitian yang dilakukan adalah pembuatan prototipe filtrasi asap industri untuk meminimalisir polusi udara bebas menggunakan mikrokontroler ESP8266. Hal ini dilakukan karena perkembangan industri sangat cepat terutama di Indonesia. Apabila perkembangan ini tidak diikuti dengan teknologi pembuangan asap yang baik maka akan menimbulkan polusi udara[2].

Menteri Perindustrian Indonesia Kabinet Kerja (2016–2019) Airlangga Hartarto pernah menyatakan bahwa pada industri semen diharapkan melakukan perubahan sesuai perkembangan teknologi saat ini di era industri 4.0 secara bertahap dengan tujuan mencari langkah efisiensi dan optimalisasi untuk mencapai hasil maksimal[3]. Dimana salah satunya adalah penerapan *IoT (Internet of Things)*.

Politeknik Negeri Jakarta (PNJ) juga sudah menerapkan IoT seperti pada penelitian yang dilakukan oleh Rifki Nur Ilham, Sonki Prasetya, dan Agus Sukandi dengan judul Sistem Monitoring Pendingin Pada Panel Surya Berbasis IoT. Penelitian ini bertujuan untuk mempermudah pemantauan panel surya yang sekarang masih banyak dilakukan secara manual. Pada penelitian tersebut mereka menggunakan kontroler ESP32 Dev Kit dan beberapa sensor seperti sensor tegangan, sensor arus, sensor suhu, sensor cahaya, sensor kecepatan angin, dan sensor kecepatan air. Nilai dari setiap sensor kemudian dikirim ke sebuah *platform IoT* yaitu Thinger.IO yang tersedia juga dalam bentuk aplikasi android maupun IOS[4].

Pada *Bag Filter* coal mill PT. SBI Tuban 2 (L62-BF1) menggunakan mekanisme purging untuk membersihkan filter atau bag dari debu kotor. Dengan memanfaatkan udara bertekanan untuk membuat purging pada filter. L62-BF1 memiliki *equipment purging* sebanyak 16 kelompok. Dimana setiap kelompok terdiri dari 8 purging, jadi total keseluruhan system purging pada *Bag Filter* tersebut sebanyak 128 unit.

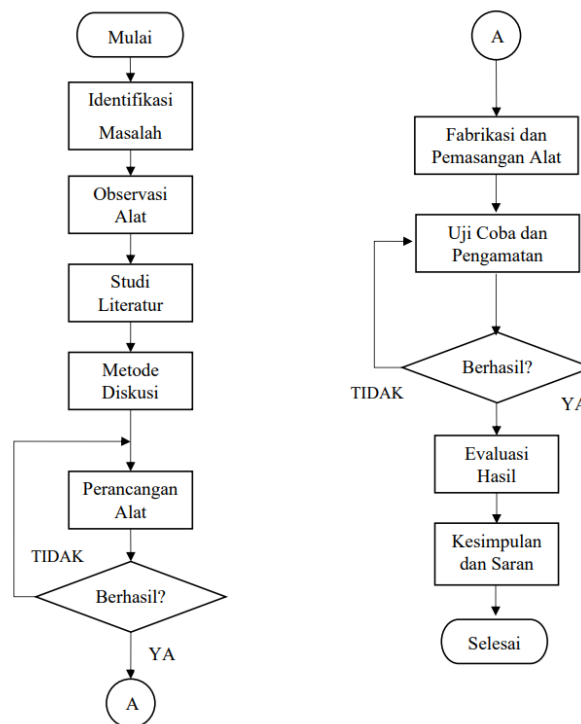
Purging pada *Bag Filter* harus di cek secara berkala di hampir setiap shift. Pengecekan *purging* pada saat ini masih dilakukan secara manual dengan mengandalkan indra pendengaran untuk mengetahui bahwa purging masih bekerja dengan baik atau tidak. Tetapi cara tersebut masih memiliki banyak kekurangan, karena pada *Bag Filter* ini memiliki 128 *purging* unit yang apabila hanya mengandalkan pendengaran saja akan membutuhkan waktu lama untuk mengetahui *purging* pada kelompok dan nomer berapa yang tidak bekerja karena purging bekerja secara acak. Dengan system yang sekarang rata-rata *patroller* memerlukan waktu 30 menit sampai 1 jam dengan kemungkinan terjadi kesalahan 2-3 kali untuk menemukan purging yang rusak.

Berdasarkan hal diatas, kami menawarkan ide untuk membuat suatu rancangan alat pemantau / monitoring purging dengan menggunakan sistem berbasis *IoT (Internet of Things)* yang dapat dilihat dan dipantau dimanapun dan kapanpun, sehingga harapannya ketika ada *purging* yang tidak bekerja dapat langsung diketahui unit yang mana dan dapat dilakukan perbaikan secepatnya.

## 2. METODOLOGI PENELITIAN

Sistem yang tertulis pada paper ini mampu mendeteksi aliran udara bertekanan yang keluar ketika aktifitas *purging* terjadi. Pada sistem ini banyaknya sensor disesuaikan dengan ketersediaan pin interrupt dan jumlah unit *purging* dalam 1 kompartemen, yaitu 8 unit. Semua data dari 8 sensor di rekam oleh mikrokontroler NodeMCU ESP8266. *Opensource cloud IoT platform* untuk menyimpan dan memvisualisasikan data dari sensor dalam representasi grafis sehingga pengguna dapat memantau data dari jarak jauh selama terdapat koneksi internet. Selain dapat melakukan *monitoring* melalui website thinger.io, monitoring juga dapat melalui aplikasi mobile thinger.io.

### A. Diagram Alir Perancangan Sistem Monitoring

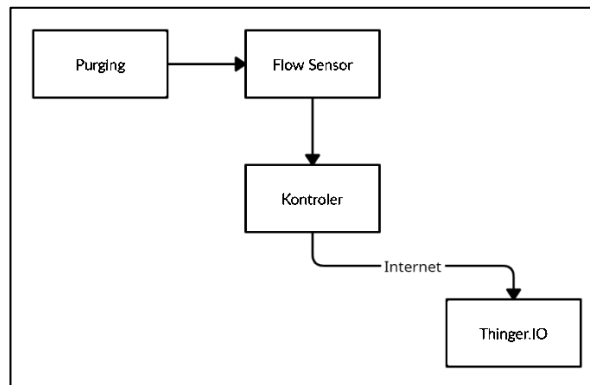


Gambar 1. Diagram Alir Perancangan

Untuk memulai pembuatan sistem *monitoring* ini penulis harus membuat diagram alir yang ditunjukkan pada Gambar 1 sebagai langkah-langkah yang akan dilakukan.

1. Identifikasi masalah dilakukan untuk mengetahui permasalahan yang terjadi, mulai dari penyebab dan dampak yang ditimbulkan.
2. Observasi alat dilakukan untuk mengetahui kondisi terbaru dari sistem purging sekarang. Dan juga melakukan pengambilan data dan informasi melalui pengukuran dan pengecekan terhadap objek-objek yang berkaitan untuk pembuatan system monitoring purging yang baru.
3. Studi literatur dengan mencari dan mempelajari informasi tentang system kerja alat, proses, dan juga teori yang terkait dengan masalah tersebut baik dari buku, jurnal, data spesifikasi, internet, maupun manual book. Jurnal-jurnal yang terkait dengan pembuatan system monitoring purging yang dilakukan.
4. Melakukkann diskusi dengan pembimbing lapangan (*Electrical Maintenance Raw Mill & Kiln Tuban 2*), *Electrical Shift*, *Patroller*, dan pihak lainnya serta Dosen untuk mendapatkan arahan serta saran dalam menangani masalah tersebut.
5. Perancangan dilakukan berdasarkan kebutuhan konsumen dan fungsi alat yang ditentukan. Mengkoneksikan NodeMCU ESP8266 ke *IoT* dilakukan agar pengiriman data dari sensor dapat dipantau melalui *platform thinger.io*. Pembuatan database dilakukan untk membuat tampilan data yang digunakan. Pengiriman data adalah mengupload program agar data dapat di visualisasikan dengan representasi grafis dalam tampilan *thinger.io*.
6. Fabrikasi alat yang telah dirancang kemudian akan dilakukan pemasangan alat pada lokasi yang sudah direncanakan.
7. Uji coba terhadap alat yang telah dibuat. Melihat fungsi alat tersebut sesuai dengan yang telah direncanakan atau tidak dan keandalan alat dalam menjalankan tugasnya dalam waktu tertentu.
8. Evaluasi hasil dilakukan untuk mengetahui apakah alat sudah beroperasi sesuai perencanaan.
9. Kesimpulan dan saran penelitian ini.

## B. Diagram Blok Sistem Monitoring

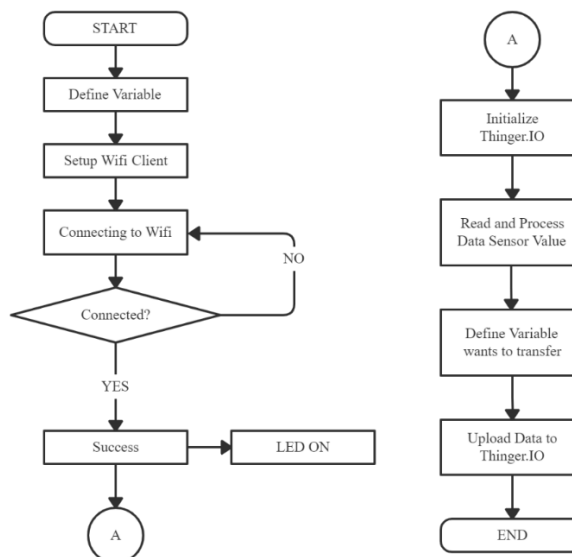


Gambar 2. Diagram Blok

Diagram blok sistem monitoring dapat dilihat pada Gambar 2. Ketika *purging* aktif maka aliran udara bertekanan yang keluar akan dibaca oleh sensor *flow* yaitu *yf-s201*. Sensor ini hanya berperan sebagai indikasi bahwa terdapat *flow* pada aliran *purging*. Ketika ada *flow* maka nilai sensor naik, sebaliknya ketika tidak ada *flow* nilai sensor akan tetap 0. Ketika tidak ada kenaikan *flow* dalam jangka waktu *sequence purging* maka unit *purging* indikasi tidak bekerja dan harus segera di cek untuk dilakukan perbaikan. Kontroler yang digunakan adalah NodeMCU ESP8266 yang berperan untuk mengumpulkan dan mengirim data dari sensor ke *thingier.io*.

## C. Diagram Alur Program

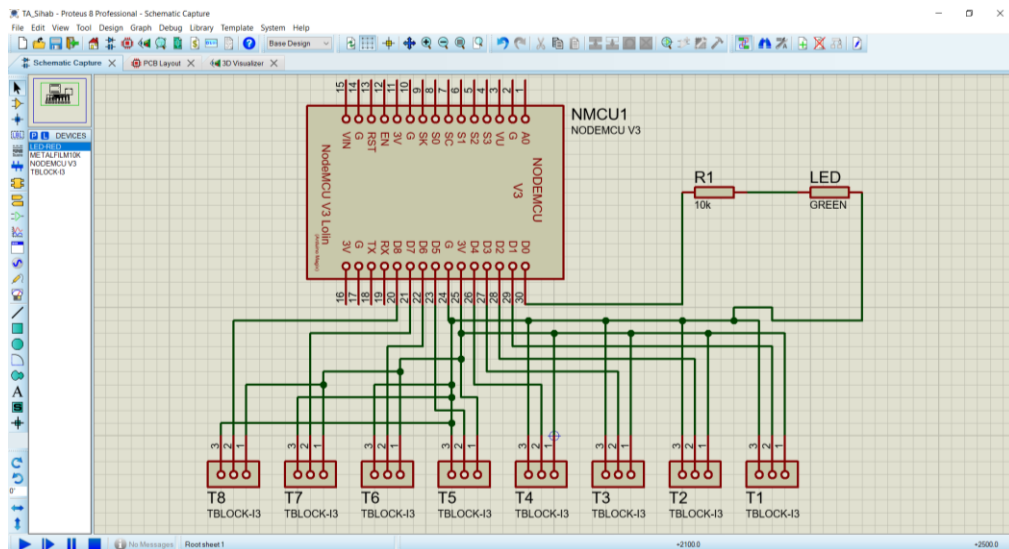
Diagram blok sistem monitoring dapat dilihat pada Gambar 3. Pertama kita buat variabel yang dibutuhkan, kemudian kita atur koneksi wifi yang akan digunakan. Ketika wifi berhasil terhubung, maka LED akan menyala sebagai indikasi bahwa kontroler berhasil terhubung dengan wifi. Selanjutnya adalah kita hubungkan kontroler ke *website*. Kita tentukan variabel mana yang akan kita kirim ke web. Data yang diterima dari sensor dan di olah oleh kontroler langsung dikirim ke web. Program selesai.



Gambar 3. Diagram Alur Program

## D. Skema Alat

Skema alat ini terdiri dari NodeMCU ESP8266, resistor, LED, dan 8 sensor *flowmeter yf-s201*. Skema alat dapat dilihat pada Gambar 4 dibawah.



Gambar 4. Skema Alat

### 3. HASIL DAN PEMBAHASAN

#### A. Fungsi *interrupt*

Penggunaan fungsi *interrupt* pada program sangat penting untuk menghindari *delay* dalam pembacaan program oleh kontroler sehingga pengiriman ke thinger.io juga tidak mengalami *delay*.

Pada penggunaan fungsi *pulseIn* untuk mengakses nilai sensor akan membuat pembacaan program mengalami *delay*. Karena fungsi *pulseIn* adalah *Blocking Function* / fungsi pemblokiran. Artinya, ketika kita atur fungsi *pulseIn* dalam kondisi *HIGH* maka fungsi tersebut akan menunggu sinyal berubah dari *LOW* ke *HIGH* dengan *timeout*. Selama waktu *timeout* tersebut eksekusi / pembacaan program akan di blokir[5]. Rata-rata *timeout* tersebut adalah 1000 ms.

Tabel 1. Kecepatan Pembacaan Program Fungsi *PulseIn*

No	Jumlah Sensor = n	Kecepatan Pembacaan Program (s)	Selisih = (n - (n - 1))
1	1	1.123	-
2	2	2.006	0.883 s
3	3	3.192	1.186 s
4	4	4.110	0.918 s
5	5	5.140	1.030 s
6	6	6.071	0.931 s
7	7	7.067	0.996 s
8	8	8.056	0.989 s
Rata-rata			0.990 s

Dari Tabel 1 dapat dilihat bahwa setiap kali ada penambahan jumlah sensor maka ada tambahan *delay* waktu pembacaan sensor juga dengan rata-rata 990 milidetik disetiap penambahannya. Apabila *delay* ini dibiarkan, maka alat yang dibuat tidak akan berjalan secara optimal.

*interrupt function* adalah fungsi untuk memprioritaskan sebuah sinyal[6]. Ketika suatu sinyal *interrupt* dipanggil maka akan langsung terbaca tanpa ada *delay*. Apabila fungsi *interrupt* sudah dijalankan, program akan langsung melanjutkan prosesnya sehingga tidak terjadi *blocking* pada pembacaan program. Dapat dilihat pada Tabel 2 yang menunjukkan waktu pembacaan program ketika menggunakan fungsi *interrupt*.

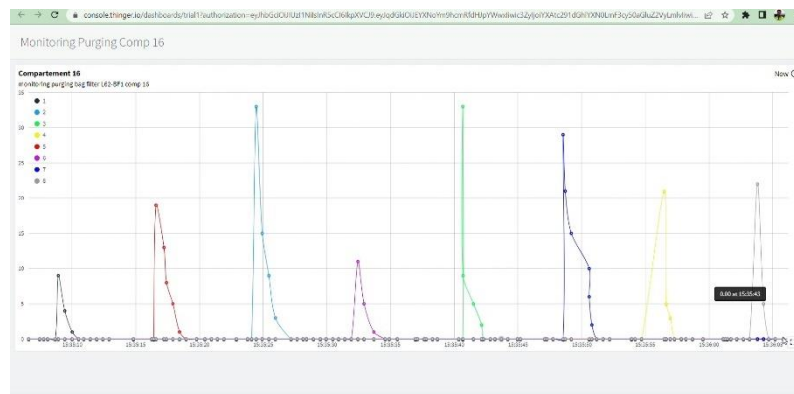
Tabel 2. Kecepatan Pembacaan Program Fungsi *Interrupt*

No	Jumlah Sensor = n	Kecepatan Pembacaan Program (s)	Selisih = (n - (n - 1))
1	1	1.117	-
2	2	1.122	0.005 s
3	3	1.098	-0.024 s

4	4	1.105	0.007 s
5	5	1.033	-0.073 s
6	6	1.112	0.079 s
7	7	1.086	-0.026 s
8	8	1.007	-0.079 s
	Rata-rata	1.085	-0.015 s

## B. Hasil Pengujian Pengiriman data ke thinger.io

Sistem *monitoring* panel *dashboard* dapat diakses melalui *website Thinger.io*. Grafik yang ditunjukkan pada gambar ada pada dashboard yang sudah kita buat dengan menggunakan widget. Tampilan *monitoring* dapat dilihat pada Gambar 4. terlihat bahwa ada kenaikan pada grafik yang menunjukkan bahwa *purging* aktif.



Gambar 5. Tampilan Antarmuka Dashboard Thinger.IO

Gambar 5 menunjukkan bagaimana tampilan antarmuka alat pada thinger.io. Pada bagian kiri tampilan terlihat keterangan untuk setiap warna berupa angka menunjukkan nomer unit *purging* nya. Nomer 1 warna hitam, nomer 2 warna biru muda, nomer 3 warna hijau, nomer 4 warna kuning, nomer 5 warna merah, nomer 6 warna ungu, nomer 7 warna biru tua, dan nomer 8 warna abu-abu.

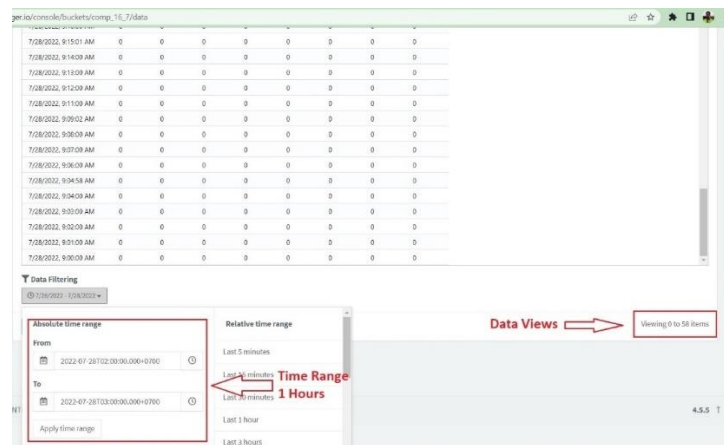
## C. Kecepatan pengiriman data

Tabel 3. Kecepatan Pengiriman Data

No	Dikirim	Diterima	Selisih (detik)
1	22:10:3	22:10:3	0
2	22:10:4	22:10:4	0
3	22:10:5	22:10:5	0
4	22:10:6	22:10:6	0
5	22:10:7	22:10:7	0
6	22:10:8	22:10:8	0
7	22:10:9	22:10:9	0
	Rata-rata		0

Selisih waktu ditampilkan pada Tabel 3. Sampling interval dan delay yang digunakan adalah 1 detik. Didapatkan rata-rata waktu transmisi adalah 0 detik.

#### D. Keandalan alat



Gambar 6. Data Bucket Thingier.IO

Dari Gambar 6 diatas terlihat bahwa pengambilan data dilakukan dengan *range* waktu selama 1 jam alat beroperasi. Idealnya ketika menggunakan waktu *sampling* 1 menit yang dilakukan selama 1 jam akan terdapat 60 data yang disimpan. Tapi pada peengujian hanya terdapat 58 data yang tersimpan, artinya ada kehilangan data sebanyak 2 data. Dari data tersebut maka dapat disimpulkan bahwa data yang tersimpan pada Thingier.IO sebesar 96.6% dari total data seharusnya.

Tabel 4. Monitoring Suhu Alat

No	Tanggal	Jam	Suhu (°C)
1	7 Juli 2022	9.00	40.0
2	7 Juli 2022	13.30	47.7
3	7 Juli 2022	16.00	44.2
4	8 Juli 2022	10.00	41.8
5	8 Juli 2022	14.00	45.1
Rata-rata			43.76

Dari Tabel 4 diatas diketahui bahwa rata-rata suhu alat dalam 2 hari pengamatan adalah 43.76 °C. Suhu mengalami kenaikan setiap menjelang siang hari, hal ini dikarenakan meningkatnya suhu pada lingkungan lokasi tugas akhir.

Suhu operasional NodeMCU ESP8266 adalah -40 °C – 125 °C[7]. Maka operasi alat pada suhu 43.76 °C masih termasuk suhu yang normal.

#### 4. KESIMPULAN

Sistem monitoring purging telah berhasil dibuat dengan rancangan optimal sistem terdiri dari NodeMCU ESP8266, *Flowmeter YF-S201*, dan website Thingier.IO dengan mempertimbangkan penggunaan fungsi interrupt, kecepatan pembacaan program 0.5 detik, dan penggunaan 8 sensor. Hasil pengujian yang dilakukan menunjukkan bahwa kecepatan yang dibutuhkan untuk mengunggah hasil pembacaan ke web Thingier.IO tergantung pada kecepatan pembacaan program dan tidak ada *delay* selama kondisi internet stabil. Dari pengujian keandalan, 96.6% data hasil pembacaan sensor terunggah ke web Thingier.IO.

#### REFERENSI

1. Dewangga, P.B., *BAG FILTER*, in *PRAMUDIA BLOG Dunia Penuh Ilmu*. 2012.
2. Manduapessy, J.O., *PERANCANGAN PROTOTIPE FILTRASI ASAP INDUSTRI UNTUK MEMINIMALISIR POLUSI UDARA BEBAS BERBASIS IoT (INTERNET OF THINGS) MENGGUNAKAN MIKROKONTROLER ESP8266*. 2020.
3. Winosa, Y. *Kemenperin Ingin Industri Semen Bertransformasi di Era 4.0*. Industri 2019 [cited 2022 14 June]; Available from: <https://www.wartaekonomi.co.id/read231250/kemenperin-ingin-industri-semen-bertransformasi-di-era-40.html>.

4. Rifki Nur Ilham, S.P., Agus Sukandi, *Sistem Monitoring Pendingin Pada Panel Surya Berbasis IoT*. 2021: p. 8.
5. Back-End, T.R. *Arduino pulseIn() function*. Arduino Tutorials [cited 2022 28 July]; Available from: [https://roboticsbackend.com/arduino-pulsein-function/#:~:text=pulseIn\(\)%20will%20block%20the,the%20duration%20of%20the%20pulse](https://roboticsbackend.com/arduino-pulsein-function/#:~:text=pulseIn()%20will%20block%20the,the%20duration%20of%20the%20pulse).
6. PAUZAN. *Interrupt pada Arduino*. 2021 [cited 2022 28 July]; Available from: <http://pauzan.com/interrupt-pada-arduino/>.
7. Make-It.ca. *NodeMCU ESP8266 Detailed Review Specifications, Overview and Setting Up Your NodeMCU*. Arduino and NodeMCU [cited 2022 28 July]; Available from: <https://www.make-it.ca/nodemcu-details-specifications/>.