

DEVELOPMENT OF APPLICATION PROGRAMMING INTERFACE (API) FOR AMIKOM PURWOKERTO HANDSANITIZER (AMPUH) DATA LOGGER VISUALIZATION

Agnis Nur Afa Zumaroh¹, Trisna Maulida², Hasri Akbar Awal Rozaq³, Rona Sepri Ananda⁴, Alif Yahya Syafa'at⁵, Imam Tahyudin*⁶, Berlilana*⁷

^{1,2,3,4,5,6,7}Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Jawa Tengah, Indonesia
Email: ¹agnisnurafaz@gmail.com, ²trisnam83@gmail.com, ³hasri150@students.amikompurwokerto.ac.id,
⁴rona123@gmail.com, ⁵alifyahya80@gmail.com, ⁶imam.tahyudin@amikompurwokerto.ac.id,
⁷berlilana@amikompurwokerto.ac.id

(Naskah masuk: 17 Maret 2022, Revisi : 27 April 2022, diterbitkan: 30 Juni 2022)

Abstract

The Internet of Things (IoT) of AMPUH adheres to three simple concepts: physical devices with IoT modules, internet-connected devices, and cloud data centers as data storage places. ThingSpeak is an IoT platform that is useful as a cloud-based data logger. Data loggers in the form of primary data or raw data need a web dashboard for data visualization because of not communicative. Therefore, this research has aim to construct API of AMPUH visualization for be used by frontend team. This research converted MATLAB programming language into PHP programming language. The data logger processing uses the powerful programming method because this method is time-efficient and can fix if an error occurs in the system development stage without having to repeat the process from the beginning. The Extreme Programming method has four steps: planning, design, coding, and testing. The processing of data loggers from the ThingSpeak platform uses the Laravel Framework to generate APIs that the frontend team will use. The researcher managed the data logger from ThingSpeak using the Laravel framework to produce several APIs used by the frontend team to visualize the data be interactive and informatively.

Keywords: Backend development, Extreme Programming, Framework, Laravel, ThingSpeak.

PEMBANGUNAN APPLICATION PROGRAMMING INTERFACE (API) UNTUK VISUALISASI DATA LOGGER AMIKOM PURWOKERTO HAND SANITIZER (AMPUH)

Abstrak

Internet of Things (IoT) AMPUH menganut tiga konsep sederhana yaitu perangkat fisik yang mempunyai modul IoT, perangkat koneksi internet dan cloud data center sebagai tempat penyimpanan data. ThingSpeak merupakan salah satu platform IoT yang berguna sebagai data logger berbasis cloud. Data logger AMPUH berbentuk data primer atau data raw yang membutuhkan adanya web dashboard untuk visualisasi data karena kurang komunikatif. Oleh karena itu, penelitian ini bertujuan untuk membangun API visualisasi data logger AMPUH untuk digunakan oleh tim frontend. Penelitian ini mengolah data logger dari bahasa MATLAB menjadi bahasa pemrograman web seperti PHP. Pengolahan data logger menggunakan metode Extreme Programming. Metode ini mempunyai waktu yang efisien dan dapat diperbaiki apabila terjadi kesalahan dalam tahap pengembangan sistem tanpa harus mengulang proses dari awal. Metode Extreme Programming memiliki empat tahapan, yaitu planning, design, coding dan testing. Pengolahan data logger dari platform ThingSpeak menggunakan Laravel Framework untuk menghasilkan API yang akan digunakan oleh tim frontend. Penelitian ini berhasil mengelola data logger dari ThingSpeak menggunakan framework Laravel sehingga menghasilkan beberapa API yang digunakan oleh tim frontend untuk memvisualisasikan data agar lebih menarik dan informatif.

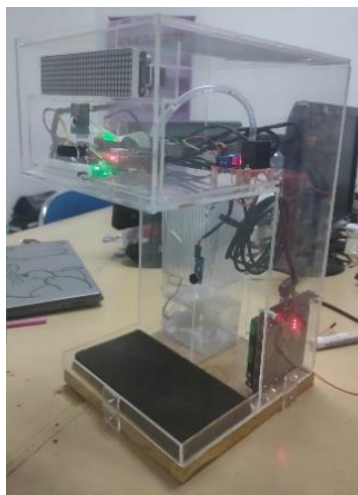
Kata kunci: Backend development, Extreme Programming, Framework, Laravel, ThingSpeak..

1. PENDAHULUAN

Kehidupan manusia sekarang nyaris tidak terlepas dengan teknologi. Semua sektor dari mulai kehidupan sehari-hari hingga kesehatan juga sudah

mengimplementasikan teknologi contohnya adalah teknologi *Internet of Things* atau biasa disebut dengan IoT. *Internet of Thing* adalah konsep untuk memanfaatkan koneksi internet yang terhubung terus-menerus[1]. Teknologi tersebut sangat mudah ditemukan seperti *smart home*, kulkas pintar, bahkan ada sepeda pintar atau biasa disebut dengan *Smart Bike* [2]. Penerapan teknologi IoT ini sangat diperlukan dibidang kesehatan terutama saat pandemi COVID-19 [3]

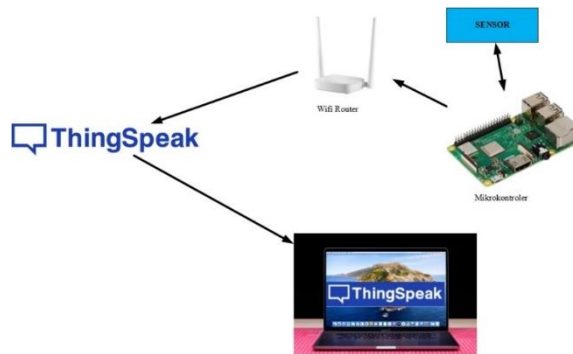
Upaya yang dilakukan oleh tim kesehatan dunia untuk meminimalisir penyebaran COVID-19 adalah dengan menggencarkan perilaku sehat yaitu dengan memakai masker, mencuci tangan pakai sabun dan menjaga jarak. Salah satu peran penerapan teknologi IoT untuk membantu penanganan COVID-19 adalah dengan cara memeriksa suhu tubuh seseorang melalui teknologi IoT [4]. Kami membangun sebuah *prototype* yang menerapkan teknologi IoT untuk membantu menanggulangi COVID-19, *prototype* tersebut bernama AMPUH (*Amikom Purwokertok Hand Sanitizer*). Ada beberapa fitur yang disediakan oleh AMPUH diantaranya pengecekan suhu dan pengecekan saturasi oksigen secara otomatis, pencatatan jumlah pengunjung, *water level indicator*, *battery management system*, perekaman data menggunakan *data logger* secara *real time*, serta *dashboard website* untuk memvisualisasi data dari data yang telah disimpan. Gambar 1 merupakan *prototype* AMPUH.



Gambar 1. *Prototype* AMPUH

Internet of Things sendiri menganut tiga konsep sederhana yaitu perangkat fisik yang mempunyai modul IoT, perangkat koneksi internet dan *cloud data center* sebagai tempat penyimpanan data [5]. Salah satu konsep IoT adalah adanya *cloud data center* sebagai tempat penyimpan data sensor[6]. *Prototype* AMPUH menggunakan *cloud computing* untuk penyimpanan *data logger*. Ada beberapa platform untuk penyimpanan data salah satunya adalah ThingSpeak. Media penyimpanan data dari *prototype* AMPUH ini menggunakan

ThingSpeak. Gambar 2 merupakan bagan alir dari proses pengiriman dan penerimaan data.



Gambar 2. Proses Pengiriman Data Sensor

Pada gambar 2, sensor yang terpasang pada AMPUH akan dikontrol langsung melalui mikrokontroler. Mikrokontroler akan dihubungkan terlebih dahulu dengan router supaya mendapatkan layanan internet. Apabila sensor bekerja, maka mikrokontroler akan mengambil data tersebut dan akan dikirim ke platform ThingSpeak melalui internet. Data yang ada pada ThingSpeak selanjutnya dapat dimonitoring melalui laptop.

ThingSpeak berperan untuk menampung data dan mengambil data kembali untuk keperluan analisis data[7]. ThingSpeak memiliki kekurangan yaitu visualisasi datanya kurang menarik, kurang komunikatif, dan kurang familiar bagi masyarakat umum. Data logger yang disediakan oleh ThingSpeak berbentuk ekstensi .csv sehingga perlu untuk diolah agar bisa dibaca oleh masyarakat umum. Gambar 3 merupakan *data logger* yang disediakan oleh ThingSpeak.

	A	B	C	D	E	F	G	H	I	J
1	created_at	entry_id	field1	field2	field3	field4	latitude	longitude	elevation	status
2	2022-01-25T16:01:29+07:00	1	36	94	6	1				
3	2022-01-25T16:01:45+07:00	2	37	90	37	1				
4	2022-01-25T16:02:54+07:00	3	35	95	4	1				
5	2022-01-25T16:06:04+07:00	4	35	92	30	1				
6	2022-01-25T16:07:11+07:00	5	36	93	88	1				
7	2022-01-26T10:54:46+07:00	6	36	95	93	1				
8	2022-01-26T10:55:55+07:00	7	37	94	24	1				
9	2022-01-26T10:58:26+07:00	8	37	94	40	1				
10	2022-01-26T11:05:07+07:00	9	38	90	69	1				
11	2022-01-26T11:10:47+07:00	10	36	94	7	1				
12	2022-01-26T11:17:47+07:00	11	35	91	56	1				
13	2022-01-26T11:48:27+07:00	12	37	95	31	1				
14	2022-01-26T11:50:28+07:00	13	38	91	30	1				
15	2022-01-26T11:57:30+07:00	14	38	91	56	1				
16	2022-01-26T12:03:23+07:00	15	36	94	28	2				
17	2022-01-17T10:55:57+07:00	16	36.63	98.36	21.50779	1				
18	2022-01-17T10:56:18+07:00	17	36.54	99.27	20.42424	1				
19	2022-01-17T10:57:03+07:00	18	36.47	98.83	20.58779	1				

Gambar 3. Data Logger dari ThingSpeak

Data mentah dari ThingSpeak perlu divisualisasikan ke dalam sebuah *dashboard website* untuk memvisualisasikan agar tampilan menjadi informatif dan mudah dipahami. Karena *dashboard website* menggunakan bahasa pemrograman web maka perlu dilakukan pengolahan *data logger* dari ThingSpeak. Pengambilan *data logger* menggunakan REST API dengan format pertukaran JSON. REST API dari ThingSpeak diolah menggunakan *library* cURL. Untuk mempermudah mengelola REST API diperlukan sebuah *framework*.

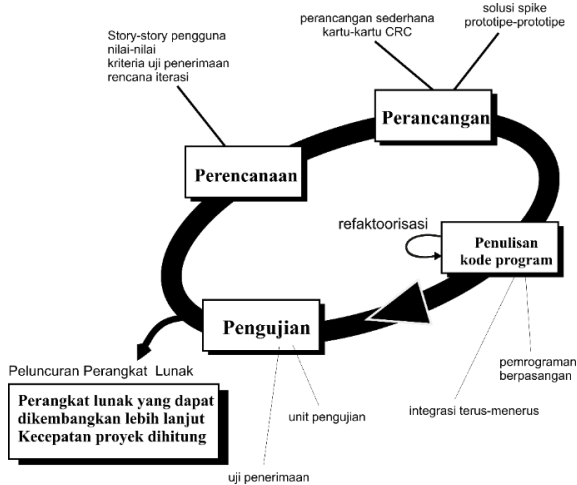
Framework adalah kumpulan *function*, *procedure*, dan *class* siap pakai untuk mempermudah

dan mempercepat pekerjaan programmer, agar programmer tidak perlu memulai dari awal[8]. Pada penelitian ini menggunakan Laravel framework dimana framework ini sudah menggunakan konsep MVC (*Model-View-Controller*) untuk membantu dalam memisahkan sistem aplikasi. MVC merupakan konsep untuk mempermudah proses penyatuan antara back-end dan front-end. MVC memecah sebuah aplikasi menjadi tiga bagian, yaitu **model** yang berhubungan dengan basis data, **view** yang berurusan dengan tampilan, dan **kontroler** yang mengatur alur jalannya aplikasi.[9][10][11].

Framework ini bersifat *open-source* atau gratis. Laravel sendiri juga menyediakan library untuk pembuatan REST API. Pengolahan cURL di dalam Laravel *framework* ini akan menghasilkan sebuah REST API. REST API tersebut mempunyai format JSON agar pengambilan data lebih mudah. Harapannya, dengan adanya penelitian ini memudahkan dalam proses pengolahan data logger.

2. METODE PENELITIAN

Metodologi yang digunakan dalam penelitian ini adalah *Software Development Life Cycle* dengan model *Extreme Programming* dapat dilihat pada gambar 4.



Gambar 4. Skema Metode *Extreme Programming*

2.1. Perencanaan

Tahap perencanaan atau *planning* dilakukan dengan mengumpulkan kebutuhan untuk memahami konteks pada pengelolaan *data logger* dan mendapatkan *output* yang dihasilkan. Pada tahap ini peneliti mengumpulkan informasi melalui observasi. Pada tahap ini menciptakan sebuah alur pengelolaan *data logger*. Kemudian peneliti memberikan nilai prioritas berdasarkan hasil *output*

2.2. Perancangan

Tahap perancangan atau *design* dilakukan dengan pembuatan perancangan sederhana mengenai

pengolahan *data logger* menggunakan *framework* Laravel. Pembuatan design ini menggunakan kartu CRC (*Class-Responsibility-Collaborator*). Dimana kartu CRC itu sendiri digunakan untuk mengidentifikasi dan mengorganisasikan *object-oriented classes*. Kartu CRC hanya men-design sistem yang dihasilkan sebagai bagian dari proses *Extreme Programming*. *Extreme Programming* mendukung adanya *refactoring* dimana *software* sistem diubah sesuai dengan kebutuhan dan hasil dari kode tidak berubah. Pada tahap ini, peneliti membuat kartu CRC untuk memprioritaskan kelas mana yang akan dikerjakan terlebih dahulu. Dalam tahap ini peneliti membuat empat kartu CRC untuk mengidentifikasi kelas yang digunakan.

2.3. Penulisan Kode Program

Pada tahap ini adalah implementasi dari tahap design yaitu membuat koding dari kartu CRC yang telah dirancang sebelumnya. Dalam pengkodean mengelola *data logger* menggunakan *framework* Laravel dengan library cURL. Pengkodean ulang dilakukan apabila hasil pengkodean ditemukan masalah atau *bugs* akan langsung dilakukan perbaikan. Pada tahapan *coding* terdapat kegiatan *pair programming* yaitu pengkodean secara berpasangan. Namun pada penelitian ini tidak dilakukan karena pada penelitian ini berfokus pada penelitian individu sehingga proses perancangan aplikasi dilakukan secara individu.

Suhu	
Suhu Tertinggi	
Suhu Terendah	
Suhu Rata-rata	
Nilai X (waktu diambilnya data suhu)	
Nilai Y (nilai suhunya)	

Pengunjung	
Jumlah Pengunjung Hari Ini	
Jumlah Pengunjung Minggu Ini	
Jumlah Pengunjung Bulan Ini	
Nilai X (menampilkan tanggal dari 7 hari kebelakang)	
Nilai Y (menampilkan jumlah pengunjung perhari dari 7 hari kebelakang)	

Saturasi Oksigen	
Saturasi Tertinggi	
Saturasi Terendah	
Saturasi Rata-rata	
Nilai X (waktu diambilnya data saturasi)	
Nilai Y (nilai saturasinya)	

Sisa Cairan Hand Sanitizer	
Sisa Cairan Hand Sanitizer	

Gambar 5. Kartu CRC

2.4. Pengujian

Bagian akhir adalah tahap pengujian terhadap pengolahan *data logger*, fokus tahapan ini adalah menguji apakah data yang dikirimkan sensor berhasil diolah oleh Laravel *framework* yang menghasilkan data berformat JSON. Metode yang digunakan adalah *black box testing* dengan melakukan pengujian terhadap masukan dan keluaran yang dihasilkan. Tahap *testing* dilakukan sampai pengujian data menghasilkan *output* JSON yang telah direncanakan pada tahap sebelumnya. Jika *output* JSON tidak sesuai maka akan dilakukan pengkodean ulang terhadap alur pengolahan data sampai pengolahan data benar-bener lulus pengujian.

3. HASIL DAN PEMBAHASAN

3.1. Perencanaan

a. Analisis Kebutuhan Fungsional

1) Kebutuhan Input

Kebutuhan masukan yang diperlukan untuk memenuhi kebutuhan dalam implementasi antara lain inputan suhu dari sensor MLX90614, inputan saturasi oksigen dari kamera raspberry pi dan inputan sisa cairan hand sanitizer dari sensor ultrasonik HC-SR04.

2) Kebutuhan Proses

Mengelola data logger dari platform ThingSpeak menggunakan bahasa pemrograman PHP dengan bantuan library cURL. agar menghasilkan sebuah API yang digunakan oleh tim frontend.

3) Kebutuhan Output

Menghasilkan sebuah API untuk menampilkan data-data yang diperlukan dalam format JSON. Data yang diperlukan antara lain data tertinggi, data terendah, data rata-rata dan data grafik berupa nilai x maupun nilai y pada data suhu dan saturasi oksigen. Kemudian data pengunjung mulai dari jumlah pengunjung hari ini, minggu ini, bulan ini dan grafik pengunjung berupa nilai x maupun nilai y. Sedangkan pada sisa cairan hand sanitizer data yang dibutuhkan adalah data terakhir pada inputan data cairan.

b. Kebutuhan Non-Fungsional

Tabel 1. Kebutuhan Non-Fungsional

<i>Hardware</i>	<i>Software</i>
Processor Intel Core i5 64bit	Sistem Operasi : Windows 11 Home Insider 64bit
RAM 8GB DDR4	Bahasa pemrograman : PHP
VGA 4GB	Text Editor : Visual Studio Code
SSD 512GB	<i>Software</i> pendukung : Postman

3.2. Perancangan

Dalam tahap design peneliti membuat kartu CRC (Class-Responsibility-Collaborator). Ada empat kartu CRC yang akan dibuat diantaranya Suhu, Saturasi Oksigen, Jumlah pengunjung dan Sisa Cairan Hand Sanitizer. Untuk keempat kartu CRC ini tidak saling berhubungan mereka berdiri sendiri untuk memuat suatu informasi. Berikut gambaran kartu CRC yang akan digunakan.

3.3. Penulisan Kode Program

a. Perisapan Implementasi

Pada penelitian ini tahap pengkodean dilakukan menggunakan framework Laravel. Laravel merupakan sebuah framework website dengan bahasa pemrograman PHP yang bersifat open-source. Framework ini diciptakan oleh Taylor Otwell, memiliki struktur Model-View-Controller

atau yang biasa disebut MVC. Ada yang berbeda dari konsep MVC milik Laravel yaitu adanya routing untuk menjembatani antara user dan controller. Sehingga controller tidak langsung menerima request dari user [12][13][14]. Sebelum melakukan instalasi Laravel diperlukan instalasi composer.

Composer merupakan fitur (dependency) tambahan PHP yang mempunyai baris seperti Command Line (CLI) dan berfungsi sebagai penginstall third-party plug-in untuk aplikasi website [15][16]. Sehingga nantinya instalasi Laravel dilakukan dengan composer.

Setelah menginstal composer, langkah selanjutnya adalah menginstall framework Laravel. Terdapat tiga cara dalam instalasi Laravel yaitu via Laravel installer, via composer dan local development server. Dalam penelitian kali ini instalasi Laravel menggunakan via composer.

Langkah pertama yang dilakukan saat instalasi Laravel adalah dengan membuka command prompt. Pastikan direktori untuk meletakkan file Laravel sudah tepat. Kelebihan dari framework Laravel yaitu tidak perlu meletakkan file Laravel di direktori xampp/htdocs, karena Laravel mempunyai server sendiri yaitu localhost:8000. Kemudian ketikkan perintah “composer create-project --prefer-dist Laravel/Laravel project_name”.

b. Implementasi Laravel Framework

Dalam mengelola data logger menggunakan frame Laravel menggunakan bantuan library milik PHP yaitu library cURL. Library ini nantinya berfungsi sebagai alat komunikasi bermacam-macam server dan jenis protocol. Untuk pengambilan data sensor dari platform ThingSpeak menggunakan API dengan protocol HTTP.

Hasil pengambilan data dari ThingSpeak melalui API yang diberikan langsung oleh platform tersebut menghasilkan data berbentuk HTML. Data HTML sulit untuk dikelompokkan jika ingin mengambil data tertentu. Dengan menggunakan Laravel maka data tersebut bisa diolah. Data bisa diolah dengan cara dikelompokkan berdasarkan field yang dibutuhkan, bisa mencari nilai maksimum, nilai minimum dan rata-rata dari sebuah field dan bisa mengelompokkan data berdasarkan “created_at”. Laravel mempunyai konsep MVC (Model-View-Controller), dalam penelitian ini peneliti hanya menggunakan controller untuk mengelola data dari ThingSpeak. Berikut ini adalah struktur framework Laravel.

Dalam mengelola data diperlukan sebuah controller, function yang ada di controller ini sudah disediakan oleh pihak Laravel sehingga peneliti bisa langsung mengelola function tersebut. Implementasi cURL digunakan untuk mengambil API dari ThingSpeak kemudian diolah menjadi array kemudian array tersebut didecode menjadi bentuk JSON. JSON akan memudahkan tim frontend dalam

menampilkan data yang untuk keperluan visualisasi data agar data tersebut menjadi informatif.

3.4. Pengujian

Data logger yang berhasil diolah kemudian di uji dengan *black box testing*. Pengujian ini bertujuan untuk menguji fungsi yang telah dibuat sebelumnya. Skenario pengujian dapat dilihat seperti pada Tabel 1. Komponen yang diuji adalah *function_suhu*, saturasi oksigen, pengujung dan sisa cairan *hand sanitizer*.

Tabel 2. Skenario Pengujian

Butir Uji	Skenario Pengujian	Jenis pengujian
Endpoint dari API Suhu	Mengirimkan route API suhu yang sudah dibuat sebelumnya menggunakan software Postman	Black Box
Endpoint dari API Saturasi Oksigen	Mengirimkan route API saturasi okesigen yang sudah dibuat sebelumnya menggunakan software Postman	Black Box
Endpoint dari API Sisa Cairan Hand sanitizer	Menghasilkan data JSON berupa status, message, nilai data sisa cairan hand sanitizer	Black Box
Endpoint dari API Jumlah Pengunjung Hari Ini	Menghasilkan data JSON berupa status, message dan data pengunjung hari ini	Black Box
Endpoint dari API Jumlah Pengunjung Minggu Ini	Mengirimkan route API jumlah pengunjung minggu ini yang sudah dibuat sebelumnya menggunakan software Postman	Black Box
Endpoint dari API Jumlah Pengunjung Bulan In	Mengirimkan route API jumlah pengunjung bulan ini yang sudah dibuat sebelumnya menggunakan software Postman	Black Box
Endpoint dari API Grafik Pengunjung	Mengirimkan route API yang menghasilkan nilai x dan nilai y, nilai x berasal dari array tanggal sedangkan nilai y berasal dari jumlah harian atau sum daily tujuh hari kebelakang yang sudah dibuat sebelumnya menggunakan software Postman	Black Box

Tabel 3. Tabel Hasil Pengujian

Butir Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
Endpoint dari API Suhu	Mengirimkan route API suhu yang sudah dibuat sebelumnya menggunakan software Postman	Menghasilkan data JSON berupa status, message, data_tertinggi, data_terendah,	[OK] Diterima [] Ditolak

Butir Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
	software Postman	rata-rata dan nilai x serta nilai y untuk grafik suhu	
Endpoint dari API Saturasi Oksigen	Mengirimkan route API saturasi okesigen yang sudah dibuat sebelumnya menggunakan software Postman	Menghasilkan data JSON berupa status, message, data_tertinggi, data_terendah, rata-rata dan nilai x serta nilai y untuk grafik suhu	[OK] Diterima [] Ditolak
Endpoint dari API Sisa Cairan Hand sanitizer	Mengirimkan route API sisa cairan hand sanitizer yang sudah dibuat sebelumnya menggunakan software Postman	Menghasilkan data JSON berupa status, message, nilai data sisa cairan hand sanitizer	[OK] Diterima [] Ditolak
Endpoint dari API Jumlah Pengunjung Hari Ini	Mengirimkan route API jumlah pengunjung hari ini yang sudah dibuat sebelumnya menggunakan software Postman	Menghasilkan data JSON berupa status, message dan data pengunjung hari ini	[OK] Diterima [] Ditolak
Endpoint dari API Jumlah Pengunjung Minggu Ini	Mengirimkan route API jumlah pengunjung minggu ini yang sudah dibuat sebelumnya menggunakan software Postman	Menghasilkan data JSON berupa status, message dan data pengunjung minggu ini yang dihitung dari tujuh hari kebelakang	[OK] Diterima [] Ditolak
Endpoint dari API Jumlah Pengunjung Bulan Ini	Mengirimkan route API jumlah pengunjung bulan ini yang sudah dibuat sebelumnya menggunakan software Postman	Menghasilkan data JSON berupa status, message dan data pengunjung bulan ini yang dihitung dari tiga puluh hari kebelakang	[OK] Diterima [] Ditolak
Endpoint dari API Grafik Pengunjung	Mengirimkan route API yang menghasilkan nilai x dan nilai y, nilai x berasal dari array tanggal sedangkan nilai y berasal dari jumlah harian atau sum daily tujuh hari kebelakang yang sudah dibuat sebelumnya menggunakan software Postman	Menghasilkan data JSON berupa nilai x dan nilai y, nilai x berasal dari array tanggal sedangkan nilai y berasal dari jumlah harian atau sum daily tujuh hari kebelakang	[OK] Diterima [] Ditolak

4. KESIMPULAN

Dari uraian pembahasan yang dilakukan pada bab sebelumnya maka bisa ditarik kesimpulan dari semua bab sebagai bahwa peneliti berhasil mengelola data logger dari ThingSpeak dengan framework Laravel sehingga menghasilkan beberapa API yang digunakan oleh tim frontend untuk memvisualisasikan data agar data tersebut bersifat informatif. Penggunaan framework Laravel dalam mengelola data logger ini cukup membantu peneliti karena di dalam framework ini menganut sistem MVC (Model-View-Controller) sehingga pengolahan data hanya berfokus pada controller. Pengujian yang dilakukan menggunakan metode black box menyatakan bahwa pengolahan data logger yang menghasilkan API bekerja dengan baik sesuai yang diharapkan. Untuk penelitian berikutnya perlu dilakukan penelitian lanjutan untuk penerapan dan analisis AMPUH pada lingkungan sebenarnya.

DAFTAR PUSTAKA

- [1] Y. Efendi, "Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile," *J. Ilm. Ilmu Komput.*, vol. 4, no. 1, pp. 19–26, 2018, doi: [10.35329/jiik.v4i1.48](https://doi.org/10.35329/jiik.v4i1.48).
- [2] M. A. Ahmed and R. Rajesh, "Implementation of Smart Refrigerator based on Internet of Things," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 2, pp. 3419–3422, 2019, doi: [10.35940/ijitee.b6343.129219](https://doi.org/10.35940/ijitee.b6343.129219).
- [3] C. Merow and M. C. Urban, "Seasonality and uncertainty in global COVID-19 growth rates," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 117, no. 44, pp. 27456–27464, 2020, doi: [10.1073/pnas.2008590117](https://doi.org/10.1073/pnas.2008590117).
- [4] M. W. Hasan, "Covid-19 fever symptom detection based on IoT cloud," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 2, pp. 1823–1829, 2021, doi: [10.11591/ijece.v11i2.pp1823-1829](https://doi.org/10.11591/ijece.v11i2.pp1823-1829).
- [5] Q. A. Putri and A. A. Slameto, "Sistem Pendeteksi Dini Kerusakan Jaringan listrik Berbasis Internet Of Think dengan Data Logger," vol. XVI, pp. 1–12, 2021.
- [6] G. Mehta, G. Mitra, and V. K. Yadav, "Application of IoT to optimize Data Center operations," in 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Sep. 2018, pp. 738–742, doi: [10.1109/GUCON.2018.8674939](https://doi.org/10.1109/GUCON.2018.8674939).
- [7] A. Marina, H. K. Ilman, F. Febi, A. E. Muhammad, and I. Muhammad, "Studi Perbandingan Platform Internet of Things (IoT) untuk Smart Home Kontrol Lampu Menggunakan NodeMCU dengan Aplikasi Web Thingspeak dan Blynk," *J. Fidel.*, vol. 2, no. 1, pp. 59–78, 2020.
- [8] T. Bin Tahir, M. Rais, and M. A. H. Sirad, "Aplikasi Point OF Sales Menggunakan Framework Laravel," *JIKO (Jurnal Inform. dan Komputer)*, vol. 2, no. 2, pp. 55–59, 2019, doi: [10.33387/jiko.v2i2.1313](https://doi.org/10.33387/jiko.v2i2.1313).
- [9] G. B. Santoso, T. M. Sinaga, and A. Zuhdi, "MVC Implementation In Laravel Framework For Development Web-Based E-Commerce Applications," *Intelmatix*, vol. 1, no. 1, pp. 37–42, 2021, doi: [10.25105/itm.v1i1.7867](https://doi.org/10.25105/itm.v1i1.7867).
- [10] T. S. Gunawan, B. H. Hayadi, C. Paramitha, M. Sadikin, "Iot Framework Current Trends and Recent Advances to Management Company in The PT.TNC," *JUDIMAS (Jurnal Pengabdian kepada Masyarakat)*, vol. 1, no. 2, pp. 164-174, 2020.
- [11] M. Ootom, N. Otoum, M. A. Alzubaidi, Y. Etoom, "An IoT-based framework for early identification and monitoring of COVID-19 cases" *J. Biomedical Signal Processing and Control*, vol. 62, no. 102149, 2020, doi: [10.1016/j.bspc.2020.102149](https://doi.org/10.1016/j.bspc.2020.102149).
- [12] A. L. Yudanto, H. Tolle, and A. H. Brata, "Rancang Bangun Aplikasi Sistem Informasi Manajemen Laboratorium Biomedik Fakultas Kedokteran Universitas Brawijaya," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 8, pp. 628–634, 2017.
- [13] E.A. Wicaksono, H. Tolle, and A. M.A.I. Pakereng, "Implementation of Laravel Framework In The Development Of Library Information System (Study Case: Smk Pgri 2 Salatiga)" *J. PILAR Nusa Mandiri*, vol. 16, no. 2, pp. 261–270, 2020, doi: [10.33480/pilar.v16i2.1666](https://doi.org/10.33480/pilar.v16i2.1666).
- [14] I. A. Tiawan, L. Afuan, "Aplikasi Pengelolaan Kerjasama Pembuatan Proyek pada Dinas Komunikasi Dan Informatika Banyumas" *JUTIF(Jurnal Teknik Informatika)*, vol. 1, no. 1, 2020, doi: [10.20884/1.jutif.2020.1.1.10](https://doi.org/10.20884/1.jutif.2020.1.1.10).
- [15] Aminudin, "Cara Efektif Belajar Framework Laravel," *Ilmu Teknol. Inf.*, vol. 1, no. 1, pp. 1–28, 2015.
- [16] M. Laaziri, K. Benmoussa, S. Khouilji, K. M. Larbi, A. El Yamami, "A comparative study of laravel and symfony PHP frameworks" *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 704-712, 2019, doi: [10.11591/ijece.v9i1.pp704-712](https://doi.org/10.11591/ijece.v9i1.pp704-712)