

Implementasi API pada Sistem Monitoring Lingkungan di Politeknik Negeri Samarinda Berbasis NodeMcu Esp8266

Alan Yuandika¹, Abdul Hamid Kurniawan², Prihadi Murdiyat³

^{1,2,3}Jurusan Teknik Elektro, Politeknik Negeri Samarinda

e-mail abdul.hamid.kurniawan@polnes.ac.id

Abstrak- Manusia hidup berdampingan dengan benda alami dan buatan serta makhluk hidup lainnya yang disebut lingkungan. Kondisi lingkungan yang nyaman dapat meningkatkan kenyamanan dalam proses belajar mengajar di lingkungan kampus Politeknik Negeri Samarinda. Sistem monitoring lingkungan kampus dibangun dengan menggunakan NodeMcu ESP8266 sebagai *sensor node* dan data pengukuran dikirimkan ke sebuah *website* agar dapat ditampilkan dan dapat diakses oleh pengguna. Data hasil pengukuran dikirimkan ke *website* dengan menggunakan API dengan beberapa variabel yang berbeda. format pengiriman data yang digunakan yaitu berbentuk JSON yang dibentuk dengan bahasa C++ untuk diupload ke NodeMcu Esp8266. pengujian data yang dilakukan hanya dengan melakukan pencocokan data sebanyak sepuluh kali pengiriman. pengujian tersebut terdiri dari pengujian dengan Postman, *sensor node* dan menampilkan *website*. Pengiriman data berhasil dilakukan dengan sepuluh kali pengiriman untuk setiap pengujian yaitu data yang dikirim sama dengan data yang diterima. Pada pengujian Postman yang dikirim dari komputer yang sama dengan yang dikirim dari komputer yang berbeda memiliki perbedaan pada *time response*. pengujian pada *sensor node* sepuluh data dapat dikirim dan diterima di server. kemudian, pada pengujian *website* mampu menampilkan data yang diterima. data dapat dikirim melalui API dan diterima di server. Dengan mengimplementasikan API pada sistem monitoring lingkungan di Politeknik Negeri Samarinda diharapkan pemantauan lingkungan dapat dilakukan secara *real-time*.

Kata kunci: API, server, *website*, monitoring, NodeMcu Esp8266, Postman, Laravel, Laravel..

I. PENDAHULUAN

Lingkungan sangat berpengaruh terhadap kelangsungan makhluk hidup yang berada di dalamnya termasuk manusia. Manusia memiliki peran besar terhadap kenyamanan dan kelestarian lingkungan. Lingkungan yang nyaman dapat menciptakan kehidupan yang sehat bagi makhluk hidup yang berada di lingkungan tersebut. Untuk Politeknik Negeri Samarinda (Polnes), kondisi lingkungan yang nyaman dapat meningkatkan kenyamanan dalam proses belajar mengajar di lingkungan kampus.

Perubahan cuaca dapat menyebabkan suhu dan kelembaban di lingkungan kampus berubah sehingga mengakibatkan lingkungan menjadi tidak nyaman. Sebagai contoh, musim panas dapat menyebabkan suasana belajar dan bekerja di Politeknik Negeri Samarinda menjadi kurang menyenangkan. Apalagi jika dilakukan penebangan pohon di area kampus. Penebangan pohon hendaknya memperhatikan

akibat yang ditimbulkannya terhadap kesehatan lingkungan di dalam kampus. Untuk itu, usaha-usaha seperti penanaman pohon, membangun atau merenovasi bangunan agar menghasilkan lingkungan yang sehat dan nyaman perlu dilakukan.

Salah satu cara yang dapat dilakukan untuk mengetahui sehat atau nyamannya lingkungan kampus adalah dengan membangun sistem monitoring lingkungan yang antara lain dapat mengukur suhu dan kelembaban udara. Pada Penelitian ini, dirancang bangun sebuah sistem monitoring lingkungan agar dapat mendeteksi suhu dan kelembaban di sekitar lingkungan kampus Politeknik Negeri Samarinda secara *real-time*.

Sistem monitoring lingkungan kampus dibangun dengan menggunakan NodeMcu ESP8266 sebagai *sensor node*. NodeMcu ESP8266 dihubungkan dengan sebuah sensor suhu dan kelembaban untuk memperoleh suhu dan kelembaban di lingkungan kampus. Kemudian, data yang diterima oleh NodeMcu ESP8266 dari sensor suhu dan kelembaban diolah untuk kemudian dikirimkan ke sebuah server agar dapat ditampilkan di sistem monitoring dan dapat diakses oleh pengguna. Dalam sistem monitoring yang dibangun tidak hanya menggunakan satu sensor node, namun lebih dari satu *sensor node*. Agar data dari berbagai *sensor node* dapat dikirim ke server maka perlu sebuah format pengiriman data atau protokol pengiriman agar mudah dikirimkan dan dapat diterima oleh server. Integrasi sebuah teknologi bernama API (*Application Programming Interface*) dipilih sebagai protokol pengiriman data dari *sensor node* ke server. API digunakan sebagai protokol pengirim data dari berbagai sensor yang mengukur nilai suhu dan kelembaban untuk monitoring lingkungan tersebut. Dengan mengimplementasikan API pada sistem monitoring lingkungan di Politeknik Negeri Samarinda diharapkan pemantauan lingkungan dapat dilakukan secara *real-time*.

Penerapan sistem monitoring suhu dan kelembaban berbasis IoT dengan API dapat meningkatkan efektivitas pemantauan dan pengambilan keputusan terkait pengelolaan lingkungan kampus secara lebih cepat dan akurat. Penelitian oleh Nugroho et al. [22] menunjukkan bahwa sistem monitoring berbasis ESP8266 dan sensor DHT22 mampu memberikan informasi lingkungan secara *real-time* dengan akurasi yang cukup baik untuk mendukung pengelolaan ruang terbuka hijau di area kampus. Selain itu, Wicaksono dan Ramadhan [23] juga menyatakan bahwa penggunaan API dalam pengiriman data

IoT dapat mempercepat proses integrasi sistem serta visualisasi data secara daring.

II. TINJAUAN PUSTAKA

A. Website

Website adalah sebuah media yang saling terhubung (*hyperlink*) dan mempunyai banyak halaman. *Website* berisi informasi yang terdiri dari teks, gambar, video, suara dan animasi. *Website* terdiri dari *website* dinamis dan *website* statis, umumnya pada saat ini *website* dinamis lebih banyak digunakan. *Website* memiliki karakteristik yaitu setiap halamannya dapat saling terhubung yang dilengkapi dengan alamat url atau *www* (*World Wide Web*) dan dapat diakses dengan jaringan internet melalui browser. Selain dapat diakses melalui internet, *website* juga dapat diakses menggunakan *localhost* tanpa internet yaitu menggunakan komputer sebagai server lokal [1]. Simbol *website* ditunjukkan pada Gambar 1.



Gambar 1. Simbol website [2]

B. Laragon

Laragon merupakan sebuah *software* yang digunakan sebagai *localhost* atau server virtual yang berjalan di sebuah laptop atau komputer. Laragon memiliki sifat yang mudah dipasang dan dioperasikan. Untuk dapat menjalankan *software* ini laptop atau komputer memiliki memori RAM minimal sebesar 4 GB. Laragon juga mendukung fitur Pretty URLs yaitu domain url yang dibuat dapat sesuai dengan keinginan pengguna [3]. *Software* Laragon dapat diunduh di laman <https://laragon.org/download/>.

C. Laravel

Laravel adalah sebuah *framework* yang digunakan untuk pembuatan atau pengembangan *website*. Laravel pertama kali diperkenalkan pada tahun 2011 oleh Taylor Otwell dengan mengusung desain MVC (*Model-View-Controller*) dan routing. Pada *framework* Laravel, akses ke sebuah *website* dijemput oleh sebuah *routing*, kemudian dari *routing* tersebut mengarah ke sebuah *controller*, sehingga untuk mengakses ke sebuah halaman *website* tidak langsung tertuju ke file halaman *website* tersebut. Laravel difasilitasi dengan fitur *php artisan* yang berfungsi untuk memudahkan pengguna melakukan pembuatan *controller* baru atau koneksi ke database baru [4]. *Framework* Laravel dapat diunduh di laman <https://laravel.com/docs/11.x/installation>.

D. API

Application programming interface (API) adalah fitur pemrograman yang dapat membuat pengguna lain memiliki akses ke sistem tanpa mengubah struktur kode utama atau basis data sistem dan membuat antar sistem dapat berkomunikasi meskipun sistem tersebut disusun dari komponen yang berbeda. API memungkinkan antar perangkat lunak dapat saling

berkomunikasi sehingga memungkinkan pengembang untuk mengakses fungsi-fungsi sistem. Ada berbagai macam jenis API, seperti API web, API perangkat keras, API sosial dan lain sebagainya [4]. Namun, API terdapat dua jenis, yaitu RESTful API dan RESTless API.

RESTful API adalah sebuah istilah yang operasinya menggunakan semua metode dari REST API seperti GET (untuk membaca), POST (untuk membuat), PUT (untuk memperbarui) dan DELETE (untuk menghapus). REST API adalah model pengembangan API untuk web yang menggunakan HTTP (*Hypertext Transfer Protocol*) untuk komunikasi dan transfer data [5]. Tujuan REST dirancang agar layanan web menjadi mudah diakses, dipahami dan diimplementasikan. REST API banyak digunakan dalam berbagai aplikasi seluler, aplikasi web dan layanan web publik lainnya [6].

Sedangkan RESTless API merupakan sebuah istilah yang dalam operasinya tidak menggunakan prinsip dari REST. Sesuai dengan istilahnya yaitu *less* yang berarti kurang dari atau tidak menggunakan prinsip dari REST. Seperti tidak menggunakan GET, POST, PUT, DELETE atau prinsip REST lainnya. Selain tidak menggunakan prinsip dari REST, beberapa RESTless juga menggunakan SOAP dan menggunakan format pengiriman data berupa XML [7].

E. Postman

Postman adalah sebuah aplikasi yang digunakan untuk merancang dan mengelola API [8]. Di dalam aplikasi Postman, pengguna dapat menguji API yang telah dibuat dengan berbagai metode yang tersedia. Dengan melakukan ujicoba melalui Postman, pengguna dapat mengetahui kemampuan API yang telah dibuat. *Software* Postman dapat diunduh di laman <https://www.postman.com/downloads/>.

F. Arduino IDE

Arduino IDE adalah sebuah *software* yang digunakan untuk memprogram board Arduino. Dengan menggunakan aplikasi ini pengguna dapat melakukan pembuatan dan pengeditan program untuk Arduino, baik Arduino Nano, Arduino Uno, Arduino Mega bahkan Esp dapat diprogram menggunakan aplikasi ini. Arduino IDE menggunakan bahasa pemrograman C++ dengan dukungan I untuk mempermudah pengguna ketika melakukan pemrograman. *Software* ini dapat diinstal pada berbagai sistem operasi komputer seperti Linux, Windows dan MAC OS [9]. *Software* Arduino IDE dapat diunduh di laman <https://www.arduino.cc/en/software>.

G. Bahasa C/C++

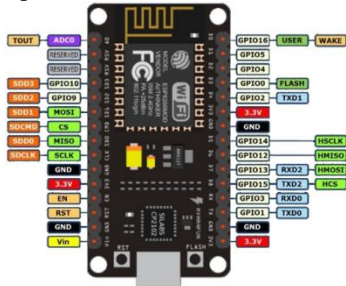
Bahasa C merupakan sebuah bahasa pemrograman yang digunakan untuk mengembangkan sebagian besar proyek open-source. Bahasa C dikembangkan sekitar tahun 1969 – 1972 oleh Dennis Ritchie dari bahasa pemrograman sebelumnya yaitu bahasa B di Bell Labs. Awalnya, bahasa C digunakan untuk menulis ulang sistem operasi UNIX. Namun, bahasa C juga dibuat sebagai bahasa pemrograman umum dan digunakan untuk membuat sistem operasi UNIX [10].

Bahasa C dan C++ memiliki perbedaan, salah satunya dalam proses penyelesaian masalah. Pada bahasa C, proses

penyelesaian dilakukan dengan membagi setiap masalah ke sub-sub masalah yang lebih kecil. Sedangkan, untuk bahasa C++ menggunakan class untuk menyelesaikan masalah [11].

H. NodeMcu Esp8266

NodeMcu Esp8266 adalah sebuah mikrokontroller yang dapat terhubung ke jaringan WiFi. NodeMcu Esp8266 dapat digunakan sebagai alat untuk kontrol dan monitoring pada sebuah proyek. NodeMcu Esp8266 memiliki fungsi yang hampir sama dengan Arduino, namun kelebihan NodeMcu Esp8266 daripada Arduino yaitu dapat terhubung ke WiFi tanpa memerlukan modul tambahan. NodeMcu Esp8266 dapat diprogram menggunakan *compiler* dari Arduino, seperti Arduino IDE dengan menggunakan port micro USB untuk melakukan *upload* program ke NodeMcu Esp8266 [12]. Gambar 2 menampilkan bentuk NodeMCU ESP8266.



Gambar 2. NodeMcu Esp8266 [13]

I. Sensor Suhu DHT 22

Sensor suhu DHT 22 merupakan sensor yang dapat mengukur suhu dan kelembaban udara. Sensor ini mirip seperti sensor suhu DHT 11, namun hasil pengukuran sensor suhu DHT 22 lebih akurat dibandingkan sensor suhu DHT 11. Output sinyal dari sensor suhu dan kelembaban DHT22 ini menggunakan sinyal output digital sehingga tidak perlu lagi mengkonversi dari sinyal analog ke sinyal digital, karena hal tersebut sudah dilakukan oleh MCU 8 bit yang ada pada modul DHT 22 tersebut. DHT 22 memiliki pengukuran jangkauan pengukuran yang lebar dan mampu mengirimkan sinyal output hingga jarak pengiriman 20 meter [14]. Gambar 3 menampilkan bentuk sensor DHT22, sedangkan Tabel 1 menunjukkan spesifikasi sensor DHT22.



Gambar 3. Sensor DHT22 [15]

TABEL 1
SPESIFIKASI DHT22 [15]

Model	DHT22
Supply voltage	3.3 – 5.5 VDC
Accuracy	±2 % RH, ±0.5 °C
Measuring range	0 - 99.9 % RH, -40 – 80 °C
Power consumption	15 µA (sleep), 1200 µA (measuring)
Sampling period	2 Second

J. Baterai

Baterai adalah suatu peralatan yang memiliki sel elektrokimia yang dapat menghasilkan tegangan untuk digunakan sebagai sumber listrik pada peralatan elektronik. Baterai diklasifikasikan ke dalam dua jenis batterai, yaitu baterai primer dan sekunder. Baterai primer adalah baterai yang hanya dapat digunakan sekali saja dan tidak dapat diisi ulang, sedangkan baterai sekunder merupakan baterai yang dapat diisi ulang. Baterai modern dikembangkan pertama kali oleh Alexandro Volta seorang fisikawan asal Italia pada abad ke 18 [16]. Gambar baterai ditunjukkan pada Gambar 4 dan spesifikasi baterai ditunjukkan pada Tabel 2.



Gambar 4. Baterai [17]

TABEL 2
SPESIFIKASI BATERAI [18]

Product Category	Lithium Ion Rechargeable Battery Cell
Nominal Capacity	2100 mAh
Rated Capacity	2000 mAh
Average Operating Voltage	3.6 V

K. Sel surya

Sel surya merupakan sebuah peralatan yang dapat mengubah energi matahari menjadi energi listrik [19]. Sel surya membangkitkan energi listrik yang berasal dari radiasi matahari. Proses pembangkitan energi listrik tersebut terjadi pada sistem fotovoltaiik. Intensitas radiasi matahari sangat mempengaruhi daya listrik yang dihasilkan. Semakin besar intensitas matahari yang diterima maka semakin besar pula daya listrik yang dihasilkan [20]. Penggunaan sel surya pada skripsi ini adalah sebagai sumber energi untuk pengecasan baterai yang digunakan. Pada Gambar 5 ditunjukkan sel surya yang digunakan.



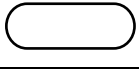
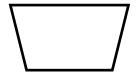
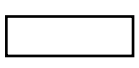
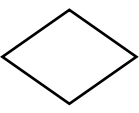
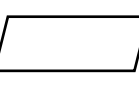
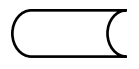
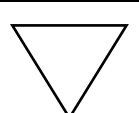
Gambar 5. Sel surya

L. Flowchart

Langkah-langkah yang dilakukan dan urutan prosedur dari suatu program dapat digambarkan dapat digambarkan dalam bentuk *flowchart*. Dengan menggunakan *flowchart*, alur suatu penyelesaian masalah tampak lebih jelas, sehingga evaluasi terhadap kegiatan yang telah dilakukan atau yang akan dilakukan lebih mudah dilakukan. *Flowchart* dapat digunakan untuk menyajikan sebuah kegiatan manual, kegiatan

pemrosesan ataupun keduanya dengan simbol-simbol *flowchart* [21] seperti tampak pada [Tabel 3](#).

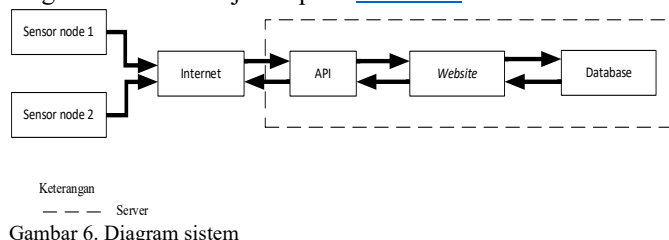
TABEL 3
SIMBOL *FLOWCHART* [21]

Simbol	Nama	Keterangan
	Arus/flow	Untuk menyatakan jalannya arus suatu proses
	Terminal	Untuk menyatakan permulaan atau akhir dari suatu proses
	Simbol Manual	Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
	Proses	Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer, biasanya menghasilkan perubahan terhadap data atau informasi
	Logika	Untuk menunjukan suatu kondisi tertentu, dengan dua kemungkinan, Ya / Tidak
	Input/output	Untuk menyatakan proses <i>input</i> dan <i>output</i> , biasanya digunakan untuk mencari catatan referensi
	Disk storage	Untuk menyatakan proses <i>input</i> yang berasal dari disk atau <i>output</i> yang disimpan ke disk
	Dokumen	Untuk mencetak dokumen
	Offline storage	Untuk menunjukan bahwa data dalam simbol ini akan disimpan ke suatu benda tertentu

III. METODE PENELITIAN

A. Diagram Sistem

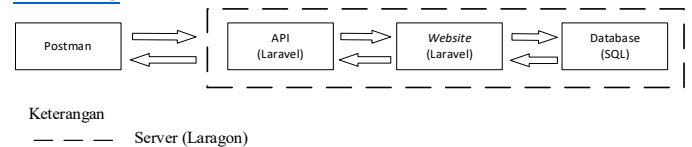
Sensor node dirancang dengan menggunakan NodeMcu Esp8266 dan DHT22 untuk dapat melakukan pengukuran suhu dan kelembaban. Data yang diukur oleh *sensor node* dikirimkan melalui API ke sebuah server untuk dapat ditampilkan di sebuah *website*. *Website* dibangun dengan menggunakan *framework* Laravel dan mysql sebagai databasenya. Untuk diagram sistem ditunjukan pada [Gambar 6](#).



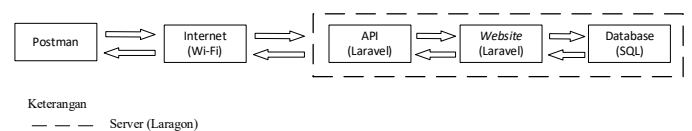
Gambar 6. Diagram sistem

B. Hubungan Antar Software yang Digunakan

Software yang digunakan memiliki hubungan yang saling berkaitan, hal ini Postman digunakan sebagai *software* pengujian API yang disimulasikan sebagai pengirim data (*sensor node*) ke server. Pengujian dengan postman dapat diakses dari dua komputer yang berbeda, yaitu dari komputer server dan dari komputer yang berbeda. Hubungan antar *software* yang diakses dari komputer server ditunjukan pada [Gambar 7](#) dan dari komputer yang berbeda ditunjukan pada [Gambar 8](#).



Gambar 7. Diagram sistem jika diakses dari komputer server



Gambar 8. Diagram sistem jika diakses dari komputer lain

C. Variabel Pengiriman Data

Data yang dikirim menggunakan variabel pengiriman data agar data yang dikirim dapat mudah untuk dikenali di server. Variabel pengiriman dirancang menggunakan 4 variabel pengiriman, yaitu *id_node*, *id_message*, suhu dan kelembaban. *id_node* digunakan sebagai pengenalan *sensor node* yang melakukan pengiriman data ke server. *id_message* digunakan sebagai penanda bahwa sensor sudah mengirimkan berapa banyak pesan dan juga digunakan sebagai deteksi jika terjadi kegagalan pengiriman data ke server. Suhu merupakan sebuah nilai yang diukur dengan menggunakan sensor DHT 22 sebagai variabel pengukuran pada sistem monitoring lingkungan di Politeknik Negeri Samarinda. Kelembaban merupakan sebuah nilai yang diukur dengan menggunakan sensor DHT 22.

D. Format Pengiriman Data

Dalam penggunaan sebuah protokol komunikasi, format pengiriman data diperlukan agar bentuk pengiriman data dari setiap *node* sama. Data hasil pengukuran digabungkan ke dalam sebuah *variabel buff* dengan bentuk *StaticJsonDocument* dengan ukuran tidak lebih dari 200 Byte. Variabel *buff* dikirim ke server dalam bentuk *key* dan *value* mengikuti format tipe data JSON. Berikut cuplikan program format pengiriman data untuk setiap *sensor node*.

```

1 StaticJsonDocument<200> buff;
2 String Json;
3 buff["id_node"] = 2;
4 buff["id_message"] = counter;
5 buff["suhu"] = round(temp * 10) / 10.0;
6 buff["kelembaban"] = round(humidity * 10) / 10.0;
7
8 serializeJson(buff, Json);
9 Serial.println(Json);
10

```



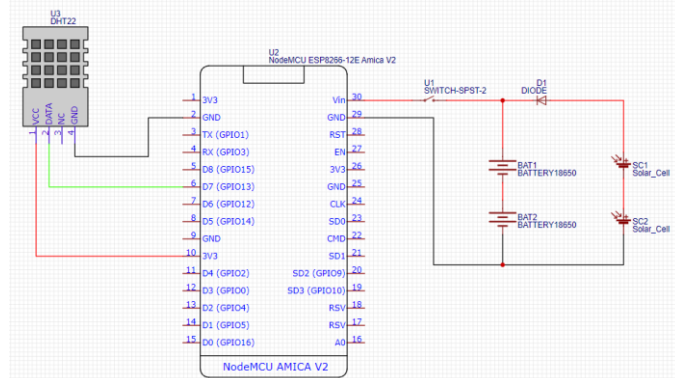
```

11 http.begin(client, url);
12 http.addHeader("Content-Type", "application/json");
13 http.POST(Json);
14 response = http.getString();
15 Serial.println(response);

```

E. Rangkaian Sensor Node

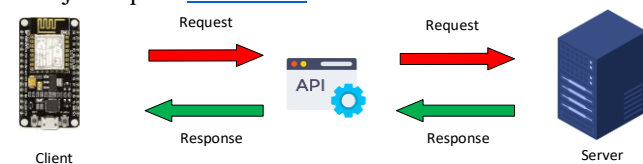
Untuk menghasilkan sebuah *sensor node* yang diharapkan, maka dilakukan pembuatan rangkaian skematik *sensor node*. Pembuatan rangkaian skematik bertujuan untuk menggambarkan koneksi antara sensor suhu DHT22 dengan NodeMcu Esp8266. Untuk setiap *sensor node* yang dibuat, digunakan model rangkaian skematik seperti pada [Gambar 9](#).



Gambar 9. Rangkaian sensor node

F. Desain API

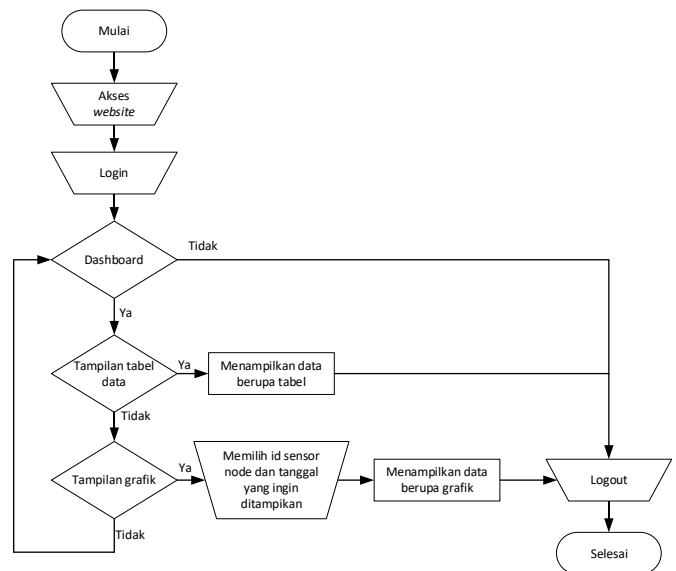
API didesain dengan menggunakan *framework* Laravel versi 6.0 dan *software* pemrograman Visual Studio Code. API yang didesain mengikuti prinsip dari REST, yaitu menggunakan metode GET dan metode POST. Desain API ditunjukkan pada [Gambar 10](#).



Gambar 10. Desain API

G. Perancangan Website

Website dirancang menggunakan *framework* Laravel dengan bantuan kode editor Visual Studio Code. *Website* yang digunakan dirancang dengan menggunakan Template AdminLTE. Template AdminLTE dapat didownload gratis di laman adminlte.io dan dapat diedit sesuai dengan kebutuhan. *Website* yang akan dirancang yaitu tampilan *dashboard*, tampilan tabel data dan tampilan grafik. Diagram alir *website* ditunjukkan pada [Gambar 11](#).



Gambar 11. Diagram alir *website*

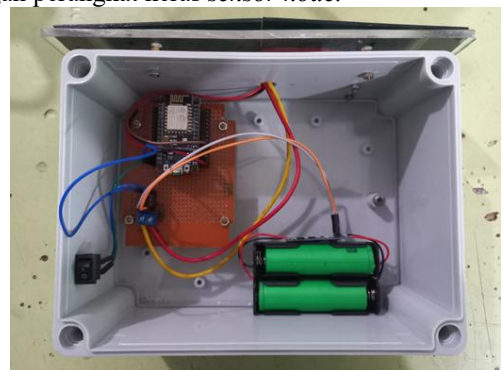
H. Metode Pengujian

Pengujian API dilakukan untuk dua metode request yang telah dirancang, yaitu metode POST dan metode GET. Untuk metode pengujian yang akan dilakukan adalah melakukan pencocokan data untuk dua metode *request* yang akan dirancang. Pada metode POST, dilakukan pengiriman data ke sebuah server untuk kemudian dicocokkan antara data yang dikirim dari client dan data yang diterima di server. Kemudian, untuk metode GET dilakukan pencocokan data antara data yang diterima di client dengan data yang tersimpan di server. Jika data pada client dan server pada dua metode yang diuji, sama maka pengujian dinyatakan berhasil.

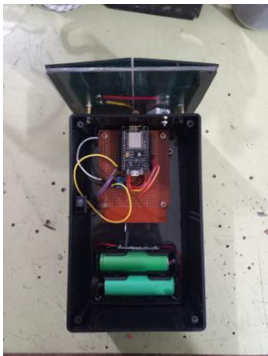
IV. HASIL DAN PEMBAHASAN

A. Hasil Perancangan dan Pengujian Perangkat Keras *Sensor node*

Sensor node dibuat di Laboratorium Mikroprocessor Jurusan Teknik Elektro dengan menggunakan NodeMcu Esp8266 dan sensor suhu DHT22. Jumlah *sensor node* yang dibuat sebanyak 2 buah dengan konfigurasi yang sama persis. Pada *sensor node*, disematkan program untuk melakukan pengiriman data ke server menggunakan API dengan metode POST. Pada [Gambar 12](#) dan [Gambar 13](#) ditunjukkan hasil perancangan perangkat keras *sensor node*.



Gambar 12. Hasil perancangan Sensor Node 1



Gambar 13. Hasil perancangan Sensor Node 2

Hasil pengujian sel surya di bawah terik matahari, nilai tegangan yang dihasilkan oleh konfigurasi seri hanya mencapai 7,6 Volt dan dihasilkan sekitar 100 mA untuk Sensor Node 1 dan sekitar 50 mA untuk Sensor Node 2. Perbedaan arus yang dihasilkan dari *mini cell* surya Sensor Node 1 dan Sensor Node 2 dapat disebabkan oleh perbedaan kemampuan sel surya dalam menangkap cahaya matahari yang diterima. Pada [Gambar 14](#), [Gambar 15](#) dan [Gambar 16](#) ditunjukkan hasil pengujian *sensor node*.



Gambar 14. Hasil pengukuran konsumsi arus *sensor node* dalam kondisi standby(kiri) dan mengirim(kanan)



Gambar 15. Hasil pengukuran tegangan yang dihasilkan oleh sel surya



Gambar 16. Hasil pengukuran arus yang dihasilkan oleh sel surya

B. Hasil Perancangan Application Programming Interface (API)

API dirancang dan diimplementasikan pada sistem monitoring lingkungan Politeknik Negeri Samarinda dengan menggunakan *framework* Laravel dan software Visual Studio Code untuk memprogramnya. API yang dirancang menggunakan dua metode, yaitu metode POST dan metode GET. Berikut cuplikan program API yang digunakan di *framework* Laravel *folder route/api.php*.

```
1 Route::post('/environment', 'EnvironmentController@create');
2 Route::get('/environment', 'EnvironmentController@baca');
```

C. Hasil Perancangan Database

Perancangan database dibuat dengan menggunakan fitur di *framework* laravel melalui terminal cmd yaitu php artisan make:model Environment -m. Database baru dengan nama Environments ditambahkan untuk menyimpan data dari sensor node dengan file model yaitu Environment.php sebagai penghubung ke database antara *framework* Laravel dengan database Environment. Di dalam database tersebut terdapat kolom berupa *id_node*, *id_message*, suhu, kelembaban dan waktu. Berikut cuplikan program untuk membuat *database* menggunakan *framework* laravel.

```
1 public function up()
2 {
3     Schema::create('environments', function (Blueprint $table) {
4         $table->bigIncrements('id');
5         $table->unsignedTinyInteger('id_node');
6         $table->unsignedTinyInteger('id_message');
7         $table->float('suhu', 3, 1);
8         $table->float('kelembaban', 3, 1);
9         $table->timestamps();
10    });
11 }
```

Agar data yang diterima di server sesuai dengan yang diharapkan, maka perlu pengaturan pembatasan data yang dapat dikirim ke server. Berikut cuplikan program yang digunakan untuk membatasi data yang dapat dikirim ke server dan pada [Gambar 17](#) merupakan bentuk database yang berhasil dibuat.

```
1 protected $fillable = [
2     'id_node',
3     'id_message',
4     'suhu',
5     'kelembaban'
6 ];
```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id	bigint(20)	UNSIGNED	Tidak	Tidak ada	Tidak ada		AUTO_INCREMENT
2	id_node	tinyint(3)	UNSIGNED	Tidak	Tidak ada	Tidak ada		
3	id_message	tinyint(3)	UNSIGNED	Tidak	Tidak ada	Tidak ada		
4	suhu	double(3,1)		Tidak	Tidak ada	Tidak ada		
5	kelembaban	double(3,1)		Tidak	Tidak ada	Tidak ada		
6	created_at	timestamp		Ya	NULL	NULL		
7	updated_at	timestamp		Ya	NULL	NULL		

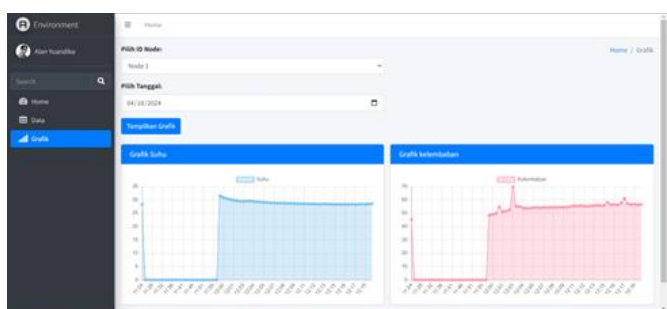
Gambar 17. Database yang dibuat

D. Hasil Perancangan Website

Website digunakan untuk menampilkan data yang diterima dari *sensor node* sehingga dapat diakses dan mudah dibaca oleh pengguna. Data hasil pengukuran dapat ditampilkan dalam dua jenis tampilan, yaitu tampilan tabel dan grafik. Hasil perancangan website ditunjukkan pada [Gambar 18](#) dan [Gambar 19](#).

No	ID	ID_node	ID_message	suhu/°C	Kelembaban/°C	Waktu
1	00	2	2	27	100	2024-09-07 09:08:05
2	00	2	2	27	100	2024-09-07 09:10:29
3	00	2	2	27	100	2024-09-09 09:06:43
4	00	2	2	27	100	2024-09-09 09:08:47
5	00	2	2	27	100	2024-09-09 09:07:09
6	00	2	2	27	100	2024-09-09 09:08:02
7	00	2	2	27	100	2024-09-09 09:04:25
8	00	2	2	27	100	2024-09-09 09:04:00

Gambar 18. Tampilan tabel data di website



Gambar 19. Tampilan grafik di website

E. Hasil Pengujian Metode POST dengan Postman dari Komputer Server

Pengujian pengiriman data dilakukan dengan menggunakan *software* Postman melalui API ke server dengan metode POST menggunakan fitur *runner* pada Postman. Pengujian dilakukan dengan mengirimkan 10 data ke server sebagai sampel pengujian. Pada [Tabel 4](#) ditunjukkan hasil pengujian sepuluh data yang dikirim ke server.

TABEL 4

HASIL PENGUJIAN METODE POST DENGAN POSTMAN YANG DIAKSES DARI KOMPUTER SERVER

Pengujian ke	Time response (ms)	Kode status	Waktu pengiriman	Waktu diterima oleh server
1	177	200 OK	23.28	23.28
2	149	200 OK	23.38	23.38
3	148	200 OK	23.48	23.48
4	204	200 OK	23.58	23.58
5	195	200 OK	00.08	00.08
6	228	200 OK	00.18	00.18
7	180	200 OK	00.28	00.28
8	191	200 OK	00.38	00.38
9	176	200 OK	00.48	00.48
10	165	200 OK	00.58	00.58

Hasil pengujian pada [Tabel 4](#) menunjukkan bahwa *time response* memiliki perbedaan untuk setiap pengirimannya. *Time response* terbesar terjadi pada pengiriman ke 6 dan terendah di pengujian ke 3. Periode waktu pengiriman yang

digunakan yaitu 10 menit. Sepuluh data yang dikirim dapat diterima dan disimpan di server.

F. Hasil Pengujian Metode GET dengan Postman dari Komputer Server

Pengujian menggunakan Postman dapat dilakukan dengan menggunakan metode GET. Postman melakukan request ke server melalui API berupa perintah untuk melakukan pengambilan data yang tersimpan di server. Tujuan pengujian ini adalah untuk memastikan bahwa data yang telah tersimpan di *database* server dapat diakses oleh perangkat lunak lain, agar dapat digunakan untuk keperluan pengembangan lainnya. Pada [Tabel 5](#) ditunjukkan hasil pengujian metode GET yang dilakukan menggunakan Postman.

TABEL 5

HASIL PENGUJIAN METODE GET DENGAN POSTMAN YANG DIAKSES DARI KOMPUTER SERVER

Pengambilan data ke	Time response (ms)	Kode status	Jumlah data diterima	Jumlah data tersimpan	Waktu diterima
1	229	200 OK	10	10	09.08
2	210	200 OK	10	10	09.18
3	173	200 OK	10	10	09.28
4	160	200 OK	10	10	09.38
5	155	200 OK	10	10	09.48
6	150	200 OK	10	10	09.58
7	148	200 OK	10	10	10.08
8	141	200 OK	10	10	10.18
9	140	200 OK	10	10	10.28
10	183	200 OK	10	10	10.38

Hasil pengujian pada [Tabel 5](#) menunjukkan bahwa ketika melakukan pengujian dengan metode GET menggunakan Postman dari komputer server diperoleh hasil yaitu *time response* memiliki nilai terbesar yaitu 229 ms dan terkecil yaitu 140 ms. Kemudian, untuk jumlah data yang tersimpan di server sebanyak 10 data dan data yang diterima juga memiliki jumlah 10 data. Untuk jumlah yang diterima di Postman sama dengan jumlah data yang disimpan di server. Sehingga, hasil pengujian dengan metode ini berhasil.

G. Hasil Pengujian Metode POST dengan Postman dari Komputer yang Berbeda

Pengiriman data menggunakan Postman dapat dilakukan dari komputer atau laptop yang berbeda dengan laptop server. Komputer yang berbeda dari komputer server dihubungkan ke jaringan Wi-Fi yang sama dengan komputer server sebagai media komunikasi yang digunakan untuk melakukan pengiriman data. Tujuan pengujian ini adalah untuk memastikan bahwa API yang dirancang mampu diakses dari perangkat lain. Pada [Tabel 6](#) hasil pengujian menggunakan Postman dari komputer yang berbeda.

TABEL 6
HASIL PENGUJIAN METODE POST DENGAN POSTMAN YANG
DIAKSES DARI KOMPUTER LAIN

Pengujian ke	Time response (ms)	Kode status	Waktu pengiriman	Waktu diterima oleh server
1	5808	200 OK	21.58	21.58
2	187	200 OK	22.08	22.08
3	184	200 OK	22.18	22.18
4	221	200 OK	22.28	22.28
5	268	200 OK	22.38	22.38
6	201	200 OK	22.48	22.48
7	215	200 OK	22.58	22.58
8	202	200 OK	23.08	23.08
9	192	200 OK	23.18	23.18
10	270	200 OK	23.28	23.28

Dari hasil pengujian yang ditunjukkan pada [Tabel 6](#) diperoleh hasil bahwa *time response* memiliki nilai terbesar pada pengujian pertama dengan nilai 5808 ms dan terkecil berada pada pengujian ke 3 yaitu sebesar 184 ms. Nilai *time response* ini memiliki nilai rata-rata yang lebih besar dibandingkan pengujian pengiriman data menggunakan Postman dari komputer server. Waktu pengiriman yang dilakukan oleh Postman memiliki waktu yang sama dengan waktu penerimaan yang tercatat di server. Dari pengujian ini, sepuluh data yang dikirim dapat diterima di server.

H. Hasil Pengujian Metode GET dengan Postman dari Komputer yang Berbeda

Pengujian mengambil data menggunakan Postman dari komputer yang berbeda juga dilakukan dengan menggunakan metode GET untuk API yang telah dibuat. Pada [Tabel 7](#) ditunjukkan hasil pengujian yang telah dilakukan.

TABEL 7
HASIL PENGUJIAN METODE GET DENGAN POSTMAN YANG
DIAKSES DARI KOMPUTER LAIN

Pengambilan data ke	Time response (ms)	Kode status	Jumlah data diterima	Jumlah data tersimpan	Waktu diterima
1	491	200 OK	471	471	23.48
2	476	200 OK	471	471	23.58
3	515	200 OK	471	471	00.08
4	5906	200 OK	471	471	00.18
5	450	200 OK	471	471	00.28
6	274	200 OK	471	471	00.38
7	494	200 OK	471	471	00.48
8	458	200 OK	471	471	00.58
9	505	200 OK	471	471	01.08
10	401	200 OK	471	471	01.18

Dari hasil pengujian pada [Tabel 7](#) diperoleh hasil bahwa nilai *time response* tercatat memiliki nilai terbesar yaitu 5906 ms pada pengujian ke 4 dan nilai terkecil yaitu 274 pada pengujian ke 6. Nilai pada pengujian ini memiliki rata-rata lebih besar dari nilai *time response* yang tercatat pada pengujian dengan metode GET dari komputer server. Jumlah data yang diterima oleh Postman sama dengan jumlah data yang tersimpan di server. Sehingga, 10 pengujian dengan cara ini seluruhnya berhasil.

A. Hasil Pengujian Pengiriman data menggunakan Sensor Node 1

Pengujian yang dilakukan dengan sensor node 1 yaitu melakukan pengiriman data melalui API ke server menggunakan jaringan Wi-Fi yang sama dengan server. Pengujian pengiriman data yang dilakukan juga hanya 10 data pengiriman, hal ini dikarenakan 10 data tersebut digunakan sebagai sampel pengujian. Pada [Tabel 8](#) ditunjukkan hasil pengujian yang telah dilakukan.

TABEL 8
HASIL PENGUJIAN PENGIRIMAN DATA MENGGUNAKAN SENSOR
NODE 1

Pengiriman Ke	Dikirim			Diterima		
	Suhu	Kelembaban	Waktu	Suhu	Kelembaban	Waktu
1	27,2	78,2	20.00	27,2	78,2	20.00
2	27,5	78,3	20.10	27,5	78,3	20.10
3	27,4	77,5	20.20	27,4	77,5	20.20
4	27,4	76,6	20.30	27,4	76,6	20.30
5	27,4	76,5	20.40	27,4	76,5	20.40
6	27,4	77,7	20.50	27,4	77,7	20.50
7	27,4	78,6	21.00	27,4	78,6	21.00
8	27,4	78,9	21.10	27,4	78,9	21.10
9	27,3	79,1	21.20	27,3	79,1	21.20
10	27,4	79,9	21.30	27,4	79,9	21.30

Dari hasil pengujian yang telah dilakukan seperti pada [Tabel 8](#) menunjukkan bahwa Sensor Node 1 dapat melakukan pengiriman data melalui API ke server. Nilai suhu dan kelembaban yang dikirim oleh Sensor Node 1 memiliki nilai yang sama dengan nilai yang diterima di server. Periode pengiriman data yang digunakan yaitu sepuluh menit. Waktu pengiriman oleh Sensor Node 1 tercatat sama dengan waktu penerimaan yang diterima di server. Hasil pengujian pengiriman data menggunakan Sensor Node 1 dianggap berhasil.

I. Hasil Pengujian Pengiriman Data menggunakan Sensor Node 2

Pengujian pengiriman data dilakukan dengan menggunakan Sensor Node 2 dan mengirimkan data pengujian ke server melalui API. Sensor Node 2 harus terhubung ke jaringan Wi-Fi yang sama dengan server agar dapat melakukan komunikasi berupa pengiriman data ke server. Pada [Tabel 9](#) ditunjukkan hasil pengujian pengiriman data menggunakan Sensor Node 2 ke server.

TABEL 9
HASIL PENGUJIAN PENGIRIMAN DATA MENGGUNAKAN SENSOR
NODE 2

Pengujian Ke	Dikirim			Diterima		
	Suhu	Kelembaban	Waktu	Suhu	Kelembaban	Waktu
1	31,3	80,2	11.08	31,3	80,2	11.08
2	31,8	79,5	11.18	31,8	79,5	11.18
3	32	79,5	11.28	32,0	79,2	11.28
4	32,3	78,2	11.38	32,3	78,2	11.38
5	32,7	77,4	11.48	32,7	77,4	11.48
6	32,9	77	11.58	32,9	77,0	11.58
7	33	78,3	12.08	33,0	78,3	12.08
8	33,1	77,2	12.18	33,1	77,2	12.18
9	33,4	75,7	12.28	33,4	75,7	12.28
10	33,6	75,4	12.38	33,6	75,4	12.38

Hasil pengujian yang ditunjukkan pada [Tabel 9](#) memiliki hasil yaitu Sensor Node 2 dapat melakukan pengiriman data ke server dengan 10 data pengujian. Nilai suhu dan kelembaban yang diterima memiliki nilai yang sama dengan nilai yang dikirim oleh Sensor Node 2. Waktu pengiriman yang tercatat di Sensor Node 2 memiliki waktu yang sama dengan waktu yang di terima di server. Sehingga, pengujian sepuluh data yang dikirim menggunakan Sensor Node 2 dinyatakan berhasil.

J. Hasil Pengujian Pengiriman Data menggunakan Sensor Node Secara Bersamaan

Pengujian pengiriman data menggunakan *sensor node* juga dilakukan pengiriman secara bersama-sama antara Sensor Node 1 dan Sensor Node 2. Sensor Node 1 dan Sensor Node 2 dinyalakan secara bersamaan untuk melakukan pengiriman data ke server. Pada [Tabel 10](#) ditunjukkan hasil pengujian pengiriman data secara bersamaan menggunakan *sensor node*.

TABEL 10
HASIL PENGUJIAN PENGIRIMAN DATA SECARA BERSAMAAN

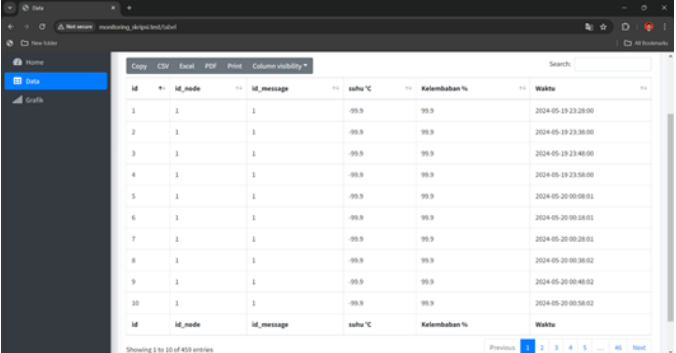
Pengiriman ke	Sensor Node 1			Sensor Node 2		
	suhu	kelembaban	waktu	suhu	kelembaban	waktu
1	27,3	65,4	17.02.28	27,5	66	17.02.25
2	27,5	66,6	17.12.28	27,6	66,8	17.12.25
3	27,4	66,5	17.22.28	27,5	66,8	17.22.25
4	27,4	66,1	17.32.29	27,4	66,6	17.32.26
5	27	67,1	17.42.29	27,3	66,7	17.42.26
6	26,6	66,9	17.52.29	27	66,3	17.52.26
7	26,4	67,5	18.02.30	26,8	66,8	18.02.27
8	26,2	68,9	18.12.30	26,7	67,7	18.12.27
9	26,1	69	18.22.30	26,6	67,8	18.22.27
10	26,1	68,8	18.32.31	26,6	67,8	18.32.28

Hasil pengujian pada [Tabel 10](#) dengan melakukan pengiriman data secara bersamaan menunjukkan hasil bahwa Sensor Node 1 dan Sensor Node 2 dapat melakukan pengiriman data secara bersamaan ke server. Waktu penerimaan yang tercatat di server antara Sensor Node 1 dan Sensor Node 2 memiliki selisih 3 detik untuk setiap pengirimannya. Hal ini dapat disebabkan oleh perbedaan waktu koneksi antara Sensor Node 1 dan Sensor Node 2. Dari pengujian tersebut, pengujian dapat dinyatakan berhasil karena sepuluh data yang dikirim oleh Sensor Node 1 dan Sensor Node 2 dapat diterima seluruhnya oleh server.

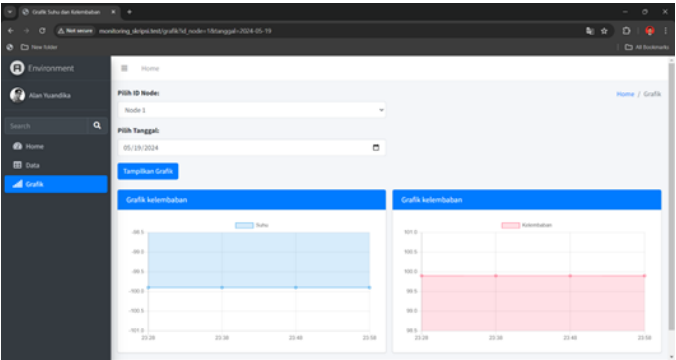
K. Hasil Pengujian menampilkan Data di Website

Pengujian menampilkan data di *website* dilakukan dengan cara mengakses *website* dan mencocokkan data suhu dan kelembaban yang tersimpan di database dengan yang ditampilkan di *website*. *Website* diakses menggunakan laptop dan *smartphone*. Namun, agar *smartphone* dapat mengakses *website* yang telah dibuat harus terhubung ke jaringan Wi-Fi yang sama dengan server. Pada [Gambar 20](#), [Gambar 21](#), dan [Gambar 22](#) ditunjukkan hasil pengujian menampilkan data di

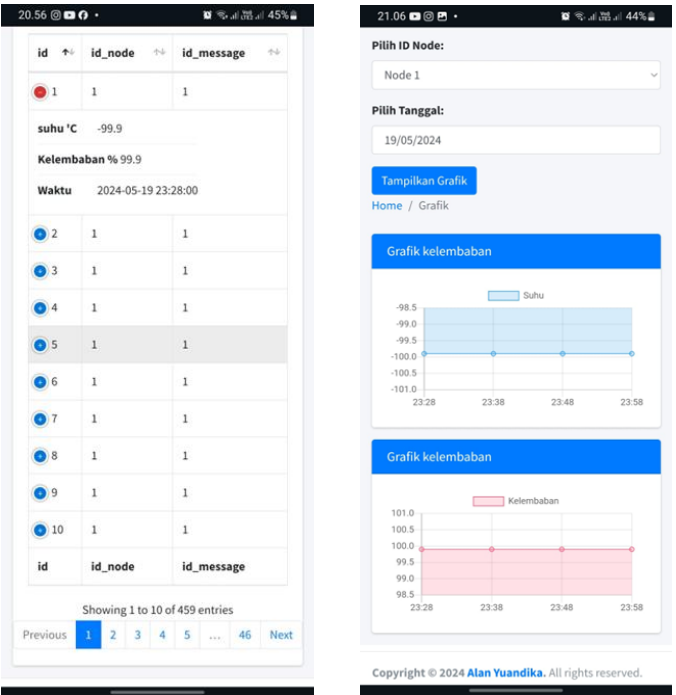
website dalam bentuk tabel dan grafik. Sedangkan, pada [Gambar 23](#) merupakan data yang tersimpan di *database* selama proses pengujian.



Gambar 20. Hasil pengujian menampilkan data dalam bentuk tabel di *website* menggunakan komputer



Gambar 21. Hasil pengujian menampilkan data menggunakan grafik di *website* menggunakan komputer



Gambar 22. Hasil pengujian menampilkan data menggunakan *smartphone* di *website* dalam bentuk tabel(kiri) dan grafik(kanan)

id	id_node	id_message	suhu	kelembaban	created_at	updated_at
1	1	1	-99.9	99.9	2024-05-19 23:28:00	2024-05-19 23:28:00
2	1	1	-99.9	99.9	2024-05-19 23:38:00	2024-05-19 23:38:00
3	1	1	-99.9	99.9	2024-05-19 23:48:00	2024-05-19 23:48:00
4	1	1	-99.9	99.9	2024-05-19 23:58:00	2024-05-19 23:58:00
5	1	1	-99.9	99.9	2024-05-20 00:08:01	2024-05-20 00:08:01
6	1	1	-99.9	99.9	2024-05-20 00:18:01	2024-05-20 00:18:01
7	1	1	-99.9	99.9	2024-05-20 00:28:01	2024-05-20 00:28:01
8	1	1	-99.9	99.9	2024-05-20 00:38:02	2024-05-20 00:38:02
9	1	1	-99.9	99.9	2024-05-20 00:48:02	2024-05-20 00:48:02
10	1	1	-99.9	99.9	2024-05-20 00:58:02	2024-05-20 00:58:02

Gambar 23. Data yang tersimpan di database

Dari hasil pengujian yang dilakukan menunjukkan bahwa *website* dapat dijalankan dari komputer server dan smartphone. Jumlah data yang digunakan untuk pengujian menampilkan data di *website* yaitu berjumlah sepuluh data. Jumlah tersebut digunakan sebagai sampel pengujian untuk pengujian menampilkan data di *website*. Kemudian, data yang ditampilkan di *website* sama dengan data yang tersimpan di database. Dari pengujian menampilkan data menggunakan *website* dinyatakan berhasil.

V. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil perancangan dan pengujian pada penelitian “Implementasi API pada Sistem Monitoring Lingkungan di Politeknik Negeri Samarinda Berbasis NodeMcu Esp8266” dapat disimpulkan bahwa:

1. API diimplementasikan pada sistem monitoring lingkungan menggunakan *framework* Laravel dan untuk menjalankan rancangan tersebut digunakan virtual server di komputer berupa *software* Laragon.
2. Tampilan data sistem monitoring lingkungan di Politeknik Negeri Samarinda menggunakan sebuah *website* dengan bantuan *framework* Laravel dan template AdminLTE sebagai desain dari tampilan *website* yang digunakan. *Website* dapat diakses melalui laptop dan perangkat yang berbeda seperti smartphone dengan koneksi jaringan yang sama.
3. Nilai *time response* tercatat memiliki hasil yang konsisten dalam 10 kali pengujian dengan menggunakan metode *request* melalui API yaitu metode POST untuk mengirim data dan metode GET untuk mengambil data dari server.
4. Pengiriman data dilakukan melalui API yang sudah dirancang dan diimplementasikan di sistem monitoring lingkungan di Politeknik Negeri Samarinda berbasis NodeMcu Esp8266 dengan sampel setiap pengujian adalah 10 kali dan semua data yang dikirim dapat diterima di server.
5. Pengiriman data melalui API ke server dapat dilakukan dengan menggunakan *software* Postman yang diakses dari komputer server dan komputer yang berbeda dari server.
6. Pengiriman data melalui API ke server dapat dilakukan dari Sensor Node 1 dan Sensor Node 2 yang terhubung ke jaringan Wi-Fi yang sama dengan server.

7. Pengiriman dari Sensor Node 1 dan Sensor Node 2 dan dari perangkat yang berbeda dengan server harus mengetahui alamat ip server yang digunakan.

B. Saran

API yang dirancang belum memiliki keamanan yang kuat untuk data yang diterima, sehingga siapapun yang mengetahui format pengiriman data melalui API yang telah dibuat maka dapat melakukan pengiriman data yang dapat mengganggu kinerja API dan server. Diharapkan untuk pengembangan berikutnya keamanan API lebih ditingkatkan.

REFERENSI

- [1] Elgamar, *Buku Ajar Konsep Dasar Pemrograman Website dengan PHP*, 1st ed. CV. Multimedia Edukasi, 2020.
- [2] Freepik, “Website symbol.” Accessed: Apr. 21, 2024. [Online]. Available: <https://www.freepik.com/vectors/website-symbol>
- [3] “Laragon.” [Online]. Available: laragon.org
- [4] H. Kurniawan et al., *Belajar Web Programming : Referensi Pengenalan Dasar Tahapan Belajar Pemrograman Web Untuk Pemula*, Jambi: PT Sonpedia Publishing Indonesia, Agustus. 2023.
- [5] B. R. Suteja dan R. Agustaf, “Interoperabilitas aplikasi berbasis Web Service,” *Jurnal Informasi Interaktif*, vol. 5, no. 3, pp. 92–147, 2020.
- [6] I. Wiguna dan I. Desnaja, *Konsep API dan Implementasinya Dalam Membangun Sistem Informasi Menggunakan Laravel*, 1st ed. Jambi: Sonpedia, 2024.
- [7] D. Gosh, “Talent500.” Accessed: Apr. 16, 2024. [Online]. Available: <https://talent500.co/blog/restful-vs-restless-api-in-backend/>
- [8] “Postman.” [Online]. Available: <https://www.postman.com/product/what-is-postman/>
- [9] J. Arifin, L. N. Zulita, dan H. Hermawansyah, “Perancangan murottal otomatis menggunakan mikrokontroler Arduino Mega 2560,” *Jurnal Media Infotama*, vol. 12, no. 1, pp. 89–98, 2016, doi: 10.37676/jmi.v12i1.276.
- [10] A. Huda, N. Ardi, dan A. Mubai, *Pengantar Coding Berbasis C/C++*, Padang: UNP Press, 2021.
- [11] H. A. Fatta, *Dasar Pemrograman C++*, Yogyakarta: CV. Andi Offset, 2006.
- [12] M. B. Ulum, Moch. Lutfi, dan A. Faizin, “Otomatisasi pompa air menggunakan Nodemcu ESP8266 berbasis internet of things (IoT),” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 1, pp. 86–93, 2022, doi: 10.36040/jati.v6i1.4583.
- [13] A. P. Manullang, Y. Saragih, dan R. Hidayat, “Implementasi Nodemcu ESP8266 dalam rancang bangun sistem keamanan sepeda motor berbasis IoT,” *JIRE (Jurnal Informatika & Rekayasa Elektronik)*, vol. 4, no. 2, pp. 163–170, 2021, [Online]. Available: <http://ejournal.stmiklombok.ac.id/index.php/jire> ISSN.2620-6900
- [14] Siswanto, I. Rojikin, dan W. Gata, “Pemanfaatan sensor suhu DHT-22, ultrasonik HC-SR04 untuk mengendalikan kolam dengan notifikasi email,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 3, pp. 544–551, 2019, doi: 10.29207/resti.v3i3.1334.
- [15] “AM2302 SIP Packaged Temperature and Humidity Sensor,” Electronic, Aosong. Accessed: May 11, 2024. [Online]. Available: <http://www.aosong.com/en/products-22.html>.
- [16] Z. Noer dan I. Dayana, *Pengantar Baterai*. Guepedia, 2021.
- [17] “The 18650 battery is the gasoline of the 21 st century,” Battery Bro. Accessed: May 28, 2024. [Online]. Available: <https://batterybro.com/>
- [18] “Cylindrical type Lithium Ion secondary battery,” Japan. Accessed: May 28, 2024. [Online]. Available: <https://www.murata.com/en-sg/products/batteries/cylindrical>
- [19] M. S. A. Amin, E. Emidiana, I. K. Pebrianti, dan Y. Irwansi, “Penggunaan panel surya sebagai pembangkit listrik pada alat pengering makanan,” *Jurnal Ampere*, vol. 7, no. 1, pp. 15–21, 2022, [Online]. Available: <https://jurnal.univpgri-palembang.ac.id/index.php/ampere/article/view/7703/5898>
- [20] A. Rachmi, B. Prakoso, H. Berchmans, I. D. Sara, and Winne, *Panduan Perencanaan dan Pemanfaatan PLTS atap di Indonesia*, AusAID, p. 94, Juni, 2020.
- [21] I. Budiman, S. Saori, R. N. Anwar, Fitriani, dan M. Y. Pangestu, “Analisis pengendalian mutu di bidang industri makanan (Studi kasus:

- UMKM Mochi Kaswari Lampion Kota Sukabumi),” *Inovasi Penelitian*, vol. 1, no. 10, pp. 2185–2190, 2021.
- [22] F. A. Nugroho, R. S. Utomo, dan I. P. Astari, “Rancang Bangun Sistem Monitoring Suhu dan Kelembaban Berbasis IoT dengan NodeMCU dan DHT22,” **Jurnal Teknik ITS**, vol. 10, no. 2, pp. C134-C138, 2021.
- [23] A. Wicaksono dan A. Ramadhan, “Implementasi API pada Sistem IoT untuk Pemantauan Lingkungan Berbasis Web,” **Jurnal Teknologi dan Sistem Komputer**, vol. 8, no. 3, pp. 200-206, 2020.