JCONES Vol. 11, No. 1 (2022) 10-20
Journal of Control and Network Systems
Situs Jurnal : http://jurnal.dinamika.ac.id/index.php/jcone

Pengembangan Dan Implementasi *Physical Web* Berbasis Rest Api Pada Sistem Pengenalan Desa Wisata

Geraldhi Aditya Putra Mahayadnya[1] **Jusak**[2] **Harianto**[3] **Pauladie Susanto**[3]
Program Studi/Jurusan Teknik Komputer
Universitas Dinamika
Jl. Raya Kedung Baruk 98 Surabaya, 60298
Email: 1)18410200053@dinamika.ac.id , 2) jusak@dinamika.ac.id, 3) harianto@dinamika.ac.id 4) pauladie@dinamika.ac.id

Abstrak: Sistem pengenalan desa wisata merupakan sebuah sistem untuk mengenalkan tempat-tempat wisata dan berbagai obyek wisata serta situs-situs sejarah di desa wisata tertentu. Penelitian untuk membuat sistem pengenalan desa wisata telah banyak diterapkan dengan teknologi berbasis web. Sistem pengenalan desa wisata sebaiknya meningkatkan *Direct Interaction*, sedangkan sistem pengenalan desa wisata berbasis web belum dapat memberikan efek *Direct Interaction* secara langsung, sehingga diperlukan sebuah sistem pengenalan desa wisata dengan tambahan efek *Direct Interaction.* Hal ini dapat dilakukan dengan *Physical Web* yang merupakan sebuah sistem untuk membagikan URL melalui *Bluetooth*. Pada penelitian ini dirancang sebuah sistem *Physical Web* yang terintegrasi dengan sistem pengenalan desa wisata berbasis web yang dijembatani oleh REST API. Hasil pengujian *Physical Web* pada tiap stasiun menunjukkan rata-rata *advertising interval* tertinggi sebesar 339 ms dengan keberhasilan pengiriman data sebesar 100% pada tiap stasiun, sedangkan pengujian *performance* pada *website* menunjukkan rata-rata skor pada tampilan *mobile* adalah 86.5 dan 86.75 pada tampilan *desktop*. Hasil pengujian *response time* tiap sistem pada server REST API menunjukkan rata-rata *response time* pada keseluruhan sistem adalah 958 ms dan *response size* pada keseluruhan sistem adalah 631 bytes. Setelah itu hasil pengujian fungsionalitas server REST API menunjukkan bahwa server REST API dapat mengolah data sebagaimana mestinya.

Kata Kunci: REST API, Desa Wisata, *Physical Web, Wordpress, Eddystone URL*

Sistem pengenalan desa wisata merupakan sebuah sistem untuk mengenalkan tempat-tempat wisata dan berbagai obyek wisata serta situs-situs sejarah di desa wisata tertentu. Para wisatawan dapat mengenal lebih dalam mengenai desa wisata tersebut dan potensi obyek wisata dengan adanya sistem pengenalan desa wisata. Sistem pengenalan desa wisata juga menjadi salah satu pendekatan untuk mempromosikan sebuah desa wisata (Rahmatillah et al., 2019). Selain itu, desa wisata yang memiliki sistem pengenalan desa wisata berpotensi menarik minat wisatawan untuk datang ke desa tersebut.

Penelitian untuk membuat sistem pengenalan desa wisata telah banyak diterapkan dengan teknologi berbasis web. Penelitian pertama telah membuat sebuah sistem pengenalan wisata berbasis web di Kabupaten Majene menggunakan Bahasa pemrograman PHP dan menggunakan XAMPP sebagai web server. Penelitian tersebut menunjukkan wisatawan mampu mengakses *website* dengan baik dan mampu mendapatkan informasi yang diinginkan (Sari & Asmawati, 2020). Kemudian penelitian kedua menerapkan sistem pengenalan wisata berbasis web dan SIG di Kabupaten Pamekasan yang menerapkan MySQL sebagai basis data pada *website* tersebut. *Website* yang telah dibuat mampu memberikan informasi dengan akurat berdasarkan *black-box testing* (Umam & Efenie, 2020). Sistem pengenalan wisata pada kedua penelitian tersebut memiliki kelebihan pada kompabilitas dengan berbagai perangkat, dan efisiensi penggunaan aplikasi, sehingga wisatawan tidak perlu memasang aplikasi tambahan. Sistem pengenalan desa wisata sebaiknya meningkatkan *Direct Interaction* dengan cara meningkatkan interaksi antara wisatawan dengan lingkungan desa wisata tersebut, sehingga potensi obyek wisata dapat lebih ditonjolkan (Pratama & Kurnia, 2018), sedangkan sistem pengenalan desa wisata berbasis web masih belum dapat memberikan efek *Direct Interaction* secara langsung terhadap

wisatawan. Berangkat dari permasalahan tersebut diperlukan sebuah sistem pengenalan desa wisata yang memungkinkan wisatawan untuk melakukan *Direct Interaction* dengan obyek wisata tersebut berbasis web.

Physical Web merupakan sebuah sistem yang memungkinkan wisatawan untuk menerima informasi berupa URL untuk mengakses *website* yang dikirmkan melalui *Bluetooth*. Sistem ini menggunakan protokol *Eddystone URL* yang beroperasi pada perangkat *Bluetooth Low Energy* (Google, 2016). Pengguna dapat mengakses sebuah *website* menggunakan URL yang telah ditransmisikan melalui *Bluetooth* dengan protokol *Eddystone URL*. Dengan penggunaaan sistem ini, maka wisatawan dapat melakukan *Direct Interaction* terhadap obyek wisata di sekitar mereka dengan sistem pengenalan desa wisata berbasis web. Implementasi *Physical Web* pada sistem pengenalan desa wisata memerlukan fitur keamanan yang mengharuskan setiap perangkat yang menggunakan *Physical Web* melakukan registrasi untuk mendapatkan URL yang valid dan dapat dikunjungi oleh wisatawan. Selain itu, perangkat-perangkat IoT yang menjalankan *Physical Web* juga memerlukan autentikasi agar integritas data yang dikirimkan terjamin. Berdasarkan permasalahan tersebut penulis menerapkan REST API sebagai penghubung antara *Physical Web* dengan sistem pengenalan desa wisata.

Pada penelitian ini dirancang sebuah sistem *Physical Web* yang terintegrasi dengan sistem pengenalan desa wisata berbasis web yang dijembatani oleh REST API. Sistem tersebut sudah banyak digunakan sebagai standar untuk pengiriman data pada aplikasi web (Mubariz et al., 2020). Pemilihan REST API didasari dari penelitian yang mengusulkan arsitektur REST dipakai pada *Web of Things* untuk *Smart City* yang memungkinkan untuk mengolah sebuah data "mentah" menjadi data yang dapat digunakan oleh sistem (Khasoggi, 2017). REST API bertugas sebagai *middleware* antara *Physical Web* dan *website*. Layanan REST API menambahkan fitur registrasi sebagai wujud autentikasi pada sistem ini. *Website* dibuat menggunakan *platform Wordpress* sebagai sistem pengenalan desa wisata. REST API bertindak sebagai penyalur informasi dan server untuk melakukan registrasi pada *Physical Web*, sedangkan *Physical Web* menyajikan informasi dari URL yang telah dikirimkan melalui protokol *Eddystone URL*. Pengguna dapat mengakses *Physical Web* melalui aplikasi Bernama "*Physical Web*" yang dapat diunduh pada *Play Store*. Perangkat yang menerapkan *Physical Web* berjumlah tiga buah dengan dua stasiun monitoring dan satu stasiun kontrol. Ketiga stasiun ini menggunakan ESP32 yang memakai *Bluetooth* dan WiFi sebagai media komunikasi.

LANDASAN TEORI

Desa Wisata

Desa wisata adalah kawasan pedesaan yang mempunyai satu atau beberapa obyek wisata yang memungkinkan untuk dikunjungi oleh wisatawan. Obyek wisata ini dapat berupa cagar alam, cagar budaya, atau tradisi khusus yang dimiliki oleh Desa tersebut. Desa wisata juga menjadi salah satu sektor pariwisata yang ada di Indonesia. Layanan pariwisata pada desa wisata biasanya berupa Paket wisata seperti akomodasi, transportasi, dan sarana wisata outdoor seperti kunjungan ke kebun dan pengenalan kebudayaan pada desa wisata (Pratama & Kurnia, 2018).

Sistem pengenalan desa wisata adalah salah satu sarana untuk mengenalkan desa wisata ke masyarakat luas. Dengan adanya sistem pengenalan wisata, informasi tentang obyek wisata yang ada pada desa wisata dapat diakses oleh para calon wisatawan untuk mencari informasi detail tentang desa wisata yang diinginkan. Sistem pengenalan desa wisata yang umum diterapkan adalah sistem pengenalan desa wisata berbasis web. Sistem wisata berbasis web ini dapat diakses secara bebas melalui internet untuk menyediakan informasi tentang desa wisata tertentu.

Physical Web

Physical Web adalah sebuah konsep yang memungkinkan pengguna perangkat *mobile* dapat berinteraksi secara langsung dengan obyek di dekatnya menggunakan web. Konsep ini dikemukakan oleh Google pada tahun 2016. *Physical Web* ini menggunakan protokol *Eddystone URL* untuk mentransmisikan URL kepada pengguna perangkat *mobile* untuk membuka *website* (Google, 2016). Konsep *Physical Web* dapat menambah unsur *discoverability* pada perangkat-perangkat IoT dan memungkinkan pengguna untuk berinteraksi melalui web, sehingga proses interaksi cukup menggunakan web browser. Sistem ini dapat dipasang dan diimplementasikan secara gratis karena *Physical Web* adalah sebuah sistem *open source*.

Saat ini *Physical Web* sudah mulai kehilangan dukungan resmi dari Google, sehingga untuk perangkat Android terbaru tidak ada dukungan terhadap *Physical Web*. Pengguna masih

dapat menggunakan aplikasi *Physical Web* melalui *Google Play Store*. Pada penelitian ini penulis menggunakan aplikasi *Physical Web* untuk berinteraksi dengan perangkat-perangkat IoT menggunakan *Bluetooth*.

REST API

Represenational State Application Programming Interface (REST API) adalah sebuah sistem yang menerapkan arsitektur antarmuka terstandardisasi untuk melayani klien melalui antarmuka yang langsung dapat digunakan tanpa memproses data yang telah diberikan terlebih dahulu. Sistem ini banyak dipakai pada server *back-end* untuk melayani jalannya aplikasi secara *online* (Hanani & Hariyadi, 2020). Aplikasi yang memakai REST API biasanya hanya menyediakan antarmuka grafis yang berinteraksi dengan klien, sedangkan pemrosesan data dan penyimpanan data dilakukan dengan REST API.

Bluetooth Low Energy (BLE)

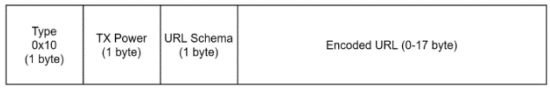
Bluetooth Low Energy (BLE) adalah sebuah sistem pengiriman data menggunakan *Bluetooth* yang memfokuskan pada transmisi data dengan daya yang rendah. BLE mampu mentransmisikan datanya dengan cara melakukan mode *sleep* selama beberapa waktu hingga proses komunikasi dimulai, sehingga mampu menghemat daya. Komunikasi menggunakan BLE memerlukan perangkat yang mendukung standar komunikasi menggunakan BLE. Perangkat ini biasa disebut dengan BLE *Beacon* (Pratiarso et al., 2018).

BLE bekerja dengan mengirimkan data kepada perangkat *mobile* yang berada dalam jangkauan BLE Beacon. Dalam proses pengiriman data menggunakan BLE, ada beberapa parameter yang terkait dengan BLE, antara lain UUID *(Universally Unique Identifier)* yang menjadi pengenal sebuah BLE Beacon, RSSI *(Received Signal Strength Indicator)*, yang menjadi ukuran kekuatan sinyal yang dipancarkan oleh BLE Beacon ketika berada pada titik jangkauan tertentu, dan *Advertising interval*, yang merupakan rentang waktu atau periode transmisi sinyal pada BLE dan diukur dengan satuan *millisecond (ms)*.

Eddystone URL

Eddystone URL adalah sebuah protokol cabang dari induk *Eddystone* yang bertugas untuk mengirimkan URL pada perangkat *Bluetooth Low Energy (BLE). Eddystone URL* bekerja dengan cara melakukan *broadcast* URL melalui perangkat BLE kepada penerima *Bluetooth* di sekitarnya (Mahayadnya et al., 2021). Ketika sebuah *Frame Eddystone URL* berhasil diterima, URL tersebut dapat dikunjungi oleh pengguna melalui aplikasi *Physical Web* atau langsung melalui *web browser*. *Eddystone URL* menjadi standar komunikasi utama bagi *Physical Web* dalam berkomunikasi dengan perangkat *mobile*.

Dalam melakukan pengiriman URL melalui protokol *Eddystone URL*, ada beberapa standar yang harus dipenuhi salah satunya adalah Panjang URL yang dikirim tidaklah lebih dari 17 *byte*. Hal ini sudah ditetapkan oleh standar protokol *Eddystone URL* itu sendiri (Google, 2017), sehingga dalam praktik pengiriman URL melalui *Eddystone URL*, URL yang dikirim disingkat terlebih dahulu menggunakan *URL Shortener*. Gambar 1 menunjukkan struktur *frame* pada *Eddystone URL*.



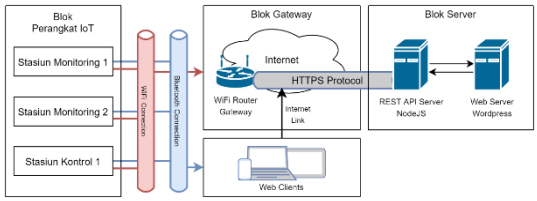
Gambar 1. Struktur frame pada *Eddystone URL*

ExpressJS

ExpressJS merupakan sebuah *library* pada NodeJS yang dipakai untuk membuat layanan REST API. ExpressJS membuat layanan API dengan protokol HTTP seperti GET, POST, PUT, dan DELETE untuk membuat *back-end* server yang mampu dipakai banyak pengguna dalam waktu bersamaan. ExpressJS mampu membuat jalur khusus atau route untuk beberapa layanan API yang berbeda. *Library* ini juga dapat dikolaborasikan dengan *library* lainnya untuk fungsional yang lebih kompleks. Sistem ini dapat dipasang dan diimplementasikan secara gratis karena ExpressJS adalah sebuah sistem *open-source* yang dapat diimplementasikan dan dikembangkan secara bebas.

METODOLOGI PENELITIAN

Model Perancangan



Gambar 2. Model perancangan sistem pengenalan desa wisata dengan *Physical Web*

Model perancangan adalah model yang menjelaskan tentang gambaran umum perancangan sistem. Gambar 2 menjelaskan tentang blok diagram perancangan. Pada penelitian ini dirancang sebuah sistem pengenalan desa wisata berbasis web yang terintegrasi dengan REST API dan *Physical Web*. Blok Perangkat IoT bertindak sebagai node yang mengakifkan layanan *Physical Web*. Blok ini mempunyai tiga stasiun yang terdiri dari dua stasiun untuk monitoring dan satu stasiun untuk kontrol. Ketiga stasiun ini berkomunikasi melalui dua media, yaitu media WiFi dan media *Bluetooth*. Media WiFi digunakan untuk berkomunikasi dengan Blok Server melalui Blok *Gateway* menggunakan protokol HTTPS, sedangkan media *Bluetooth* digunakan untuk berkomunikasi dengan web clients dengan cara mentransmisikan URL menggunakan protokol *Eddystone URL* untuk proses *Physical Web*. Pada Blok Server terdapat dua layanan server yaitu server REST API, yang merupakan server untuk memproses layanan dari Blok Perangkat IoT dan web server yang bertugas sebagai penyedia konten pengenalan desa wisata yang terpasang menggunakan *wordpress.*

Perancangan Prototipe

Model perancangan adalah model yang menjelaskan tentang gambaran umum perancangan sistem. Gambar 2 menjelaskan tentang blok diagram perancangan.

Gambar 3. Perancangan prototipe stasiun monitoring 1

Gambar 3 menunjukkan hasil perancangan prototipe stasiun monitoring 1. Terdapat sensor kelembapan DHT11 dan sensor suhu LM35 yang terpasang di sisi atas prototipe, sedangkan di sisi kanan terpasang dua lampu LED berwarna merah dan hijau untuk menunjukkan indikator dari setiap sensor. Setelah itu pada sisi bawah stasiun monitoring 1 terdapat dua buah tombol yang berfungsi untuk mengaktifkan fungsi registrasi dan unregistrasi pada stasiun monitoring 1.

Gambar 4. Perancangan prototipe stasiun monitoring 2

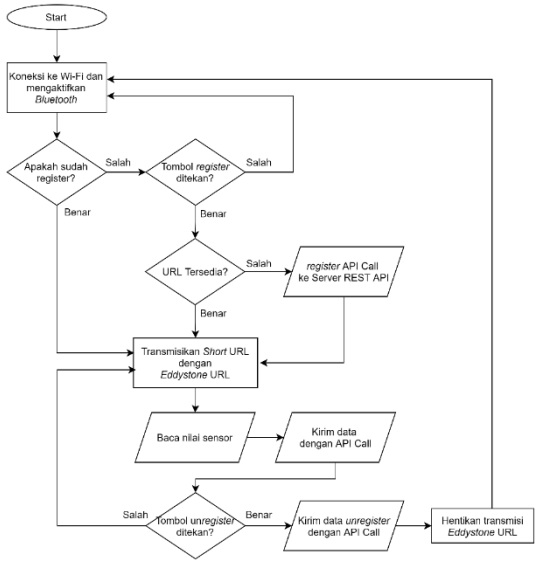
Gambar 4 menunjukkan hasil perancangan prototipe stasiun monitoring 2. Di sisi kanan terpasang dua lampu LED berwarna merah dan hijau untuk menunjukkan indikator dari setiap sensor. Setelah itu pada sisi bawah stasiun monitoring 2 terdapat dua buah tombol yang berfungsi untuk mengaktifkan fungsi registrasi dan unregistrasi pada stasiun monitoring 2, sedangkan pada bagian atas dari kotak plastik terpasang sensor intensitas cahaya LDR. Tujuan penempatan sensor pada bagian atas adalah untuk menangkap perubahan intensitas cahaya pada tempat yang terkena sinar matahari.

Gambar 5. Perancangan prototipe stasiun kontrol 1

Gambar 5 menunjukkan hasil perancangan prototipe stasiun kontrol 1. Di sisi kanan terpasang dua lampu LED berwarna merah dan hijau untuk menunjukkan indikator dari setiap sensor. Setelah itu pada sisi bawah stasiun kontrol 1 terdapat dua buah tombol yang berfungsi untuk mengaktifkan fungsi registrasi dan unregistrasi pada stasiun 1. Pada perancangan prototipe ini Relay 5V empat kanal dipasang pada bagian atas. Tujuan dari pemasangan di belakang adalah untuk menyediakan akses bagi aktuator untuk dipasang secara langsung kepada Relay 5V empat kanal.

Perancangan Perangkat Lunak

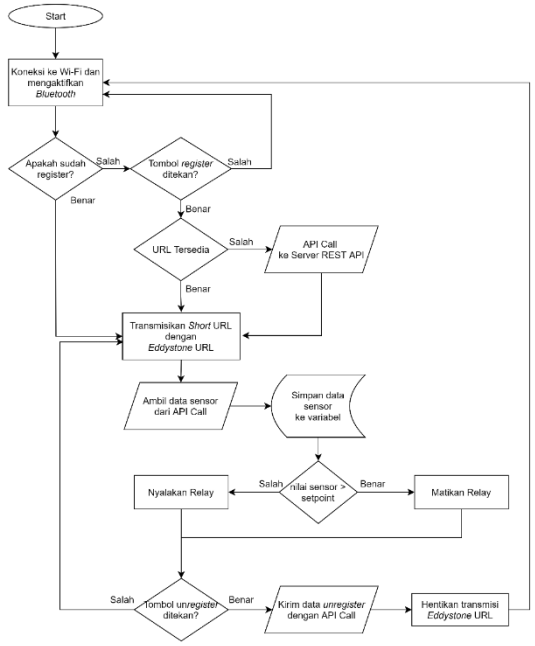
Perancangan Perangkat Lunak tiap Stasiun



Gambar 6. Diagram alir pada stasiun monitoring 1 dan 2

Gambar 6 menunjukkan diagram alir yang diterapkan dalam memprogram mikrokontroler ESP32 pada stasiun monitoring 1 dan 2. Algoritma dimulai dari membaca tombol registrasi yang terpasang pada prototipe, kemudian melakukan proses registrasi pada server REST API dan mengirimkan data melalui *Eddystone URL*. Setelah itu algoritma membaca nilai sensor dan mengirimkan data nilai sensor kepada server REST API. Jika saat pembacaan algoritma pengguna melakukan unregistrasi, maka pengguna menekan tombol unregister dan algoritma melakukan *API Call* dengan maksud melakukan unregistrasi pada server REST API. Setelah itu proses mengulang kembali menuju proses transmisi URL menggunakan *Eddystone URL*.

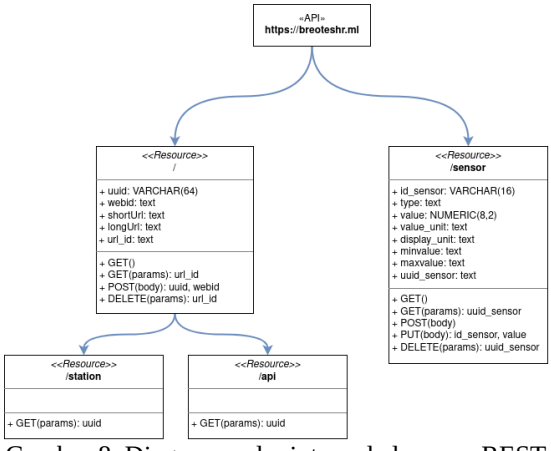
Gambar 7 menunjukkan diagram alir yang diterapkan dalam memprogram mikrokontroler ESP32 pada stasiun kontrol. Algoritma dimulai dari membaca tombol registrasi yang terpasang pada prototipe, kemudian melakukan proses registrasi pada server REST API dan mengirimkan data melalui *Eddystone URL*. Setelah itu algoritma membaca nilai sensor melalui server REST API. Kemudian algoritma melakukan penggerakan aktuator dengan sistem kontrol ON-OFF. Jika saat algoritma penggerakan aktuator pengguna melakukan unregistrasi, maka pengguna menekan tombol unregister dan algoritma melakukan *API Call* dengan maksud melakukan unregistrasi pada server REST API. Setelah itu proses mengulang kembali menuju proses transmisi URL menggunakan *Eddystone URL*.



Gambar 7. Diagram alir pada stasiun kontrol 1

Perancangan Perangkat Lunak server REST API

Perancangan perangkat lunak pada layanan REST API meliputi beberapa layanan, antara lain sistem registrasi, sistem pengiriman data sensor, sistem pembacaan data sensor, dan sistem unregistrasi. Server REST API dibuat dengan bahasa pemrograman *Javascript* dengan menggunakan NodeJS. Server ini memakai *library* ExpressJS untuk membuat layanan REST API.



Gambar 8. Diagram endpoints pada layanan REST API

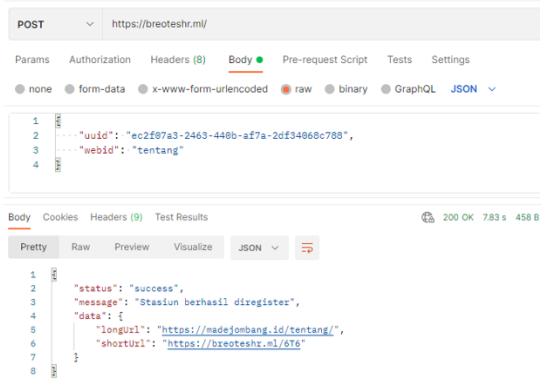
Berdasarkan gambar 8, terdapat dua *endpoints* yang berbeda untuk setiap layanan yang ada. *Endpoint* pertama (“/”) bertugas sebagai layanan registrasi pada setiap stasiun, sedangkan *Endpoint* kedua (“/sensor”) bertugas untuk menjalankan fungsi penyimpanan dan pembacaan sensor pada stasiun monitoring. Metode HTTP yang terdapat pada kedua *Endpoints* ini antara lain GET, POST, PUT, dan DELETE.

Pemanggilan API pada kedua *endpoints* ini dibedakan menjadi tiga cara. Cara pertama adalah memakai parameter untuk memanggil API. Kemudian cara kedua adalah memakai *body* untuk memanggil API. Setelah itu cara ketiga adalah tanpa memakai parameter dan *body*.

HASIL DAN PEMBAHASAN

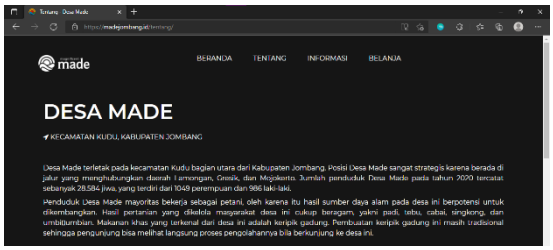
Pengujian fungsionalitas server REST API

Pengujian ini melakukan registrasi pada server dengan mengirimkan UUID dan Web ID. Web ID di sini menunjukkan kata kunci pencarian pada server layanan artikel *wordpress* yang merujuk kepada URL dari server layanan artikel *wordpress*. Pesan balasan dari server berupa URLID dan URL yang sudah disingkatkan untuk selanjutnya dipasangkan pada stasiun yang melakukan proses registrasi. Gambar 9 menunjukkan pesan permintaan dan pesan balasan dari proses registrasi ini.



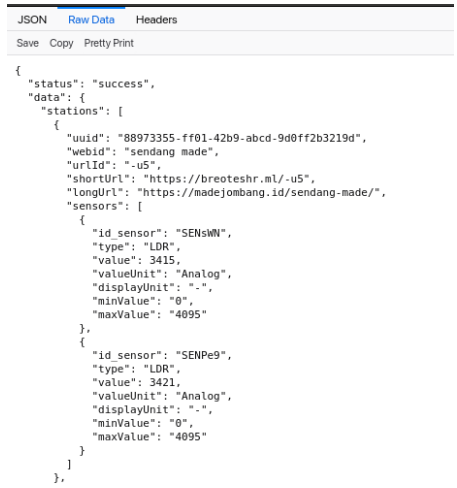
Gambar 9. Proses registrasi pada tiap stasiun dengan *API call*

Selanjutnya pengujian ini menguji sistem pengalihan dari URL pendek menjadi URL Panjang. Metode ini berjalan dengan cara melakukan GET *request* langsung ke server dengan URL (“/{URLID}”). Setelah itu server mengalihkan langsung URL yang dikirimkan menuju URL panjang hasil dari pencarian Web ID pada poin A. Gambar 10 menunjukkan respons server terhadap data yang telah dikirim.



Gambar 10. Tampilan web yang telah dialihkan dari server REST API

Setelah itu pengujian ini juga mencari seluruh stasiun yang telah registrasi ke server. Pengujian ini dilakukan dengan cara melakukan GET *request* langsung ke server dengan URL *root* (“/”). Setelah itu server memberikan respons berupa pesan yang berformat JSON. Gambar 11 menunjukkan respons server terhadap data yang telah dikirim.



Gambar 11. Respons balasan server pada metode GET Sistem Registrasi

Pengujian *Response time* Server REST API

Pengujian *response time* server REST API menguji waktu respon server REST API dalam membalas pesan dari klien, dalam hal ini adalah stasiun pada Blok Perangkat IoT. Pengujian tersebut berfungsi untuk mengetahui durasi lama server REST API memproses dan mengirimkan data kepada klien. Pengujian ini menggunakan aplikasi *Postman*. Pengujian ini menjadi penentu apakah sistem pengenalan desa wisata berbasis web dapat terintegrasi dengan REST API dan menyajikan konten kepada web clients.

Pengujian Response time pada Sistem Registrasi

Pengujian ini dilakukan dengan cara mengirimkan data kepada server REST API pada bagian sistem registrasi dengan metode acak sebanyak 10 kali. Setelah itu dilakukan analisis terhadap data yang telah dikirimkan. Hal ini bertujuan untuk mengetahui *response time* terhadap sistem registrasi.

Tabel 1. Pengujian *response time* pada sistem registrasi

No	Metode	*Response time* (ms)	*Response size* (byte)
1	POST	7090	458
2	POST	2540	458
3	POST	3760	458
4	GET	370	503
5	GET	1436	503
6	GET	479	367
7	GET	900	367
8	DELETE	469	368
9	DELETE	960	368
10	DELETE	2560	368
Rata-rata		2056	422

Berdasarkan tabel 1, terdapat 10 data uji dengan metode GET, POST, dan DELETE. Rata-rata *response time* pada sistem registrasi adalah 2056 ms, sedangkan rata-rata dari *response size* pada sistem registrasi adalah 422 *bytes*. *Response time* pada sistem registrasi dipengaruhi oleh koneksi internet dari stasiun ke internet dan koneksi dari server REST API ke server layanan artikel *wordpress.*

Pengujian Response time pada Sistem Pengiriman Data Sensor

Pengujian ini dilakukan dengan cara mengirimkan data kepada server REST API pada bagian sistem pengiriman data sensor dengan metode acak sebanyak 10 kali. Setelah itu dilakukan analisis terhadap data yang telah dikirimkan. Hal ini bertujuan untuk mengetahui *response time* terhadap sistem pengiriman data sensor.

Tabel 2. Pengujian respose time pada sistem pengiriman data sensor

No	Metode	*Response time* (ms)	*Response size* (byte)
1	POST	113	315
2	POST	245	315
3	POST	901	633
4	POST	873	633
5	POST	563	633
6	PUT	881	421
7	PUT	267	421
8	PUT	653	421
9	PUT	147	455
10	PUT	221	455
Rata-rata		480	470

Berdasarkan tabel 2, terdapat 10 data uji dengan metode POST dan PUT. Rata-rata *response time* pada sistem pengiriman data sensor adalah 486 ms, sedangkan rata-rata dari *response size* pada sistem pengiriman data sensor adalah 470 bytes. *Response time* pada sistem pengiriman data sensor dipengaruhi oleh koneksi dari stasiun ke server REST API melalui internet.

Pengujian Response time pada Sistem Pembacaan Data Sensor

Pengujian ini dilakukan dengan cara mengirimkan data kepada server REST API pada bagian sistem pembacaan data sensor dengan metode acak sebanyak 10 kali. Setelah itu dilakukan analisis terhadap data yang telah dikirimkan. Hal ini bertujuan untuk mengetahui *response time* terhadap sistem pembacaan data sensor.

Tabel 3. Pengujian *response time* pada sistem pembacaan data sensor

No	Metode	*Response time* (ms)	*Response size* (byte)
1	GET	581	471
2	GET	767	471
3	GET	301	471
4	GET	869	614
5	GET	399	614
6	GET	818	1980
7	GET	254	1980
8	GET	522	1980
9	GET	391	1980
10	GET	720	1980
Rata-rata		562	1254

Berdasarkan tabel 3, terdapat 10 data uji dengan metode GET. Rata-rata *response time* pada sistem pembacaan data sensor adalah 562 ms, sedangkan rata-rata dari *response size* pada sistem pembacaan data sensor adalah 1254 bytes. *Response time* pada sistem pembacaan data sensor dipengaruhi oleh koneksi dari stasiun ke server REST API melalui internet.

Pengujian Response time pada Sistem Unregistrasi

Pengujian ini dilakukan dengan cara mengirimkan data kepada server REST API pada bagian sistem unregistrasi dengan metode acak sebanyak 10 kali. Setelah itu dilakukan analisis terhadap data yang telah dikirimkan. Hal ini bertujuan untuk mengetahui *response time* terhadap sistem unregistrasi.

Tabel 4. Pengujian *response time* pada sistem unregistrasi

No	Metode	*Response time* (ms)	*Response size* (byte)
1	DELETE	1165	386
2	DELETE	696	364
3	DELETE	431	364
4	DELETE	915	364
5	DELETE	381	364
6	DELETE	480	386
7	DELETE	1029	386
8	DELETE	923	386
9	DELETE	430	386
10	DELETE	808	386
Rata-rata		726	377

Berdasarkan tabel 4, terdapat 10 data uji dengan metode DELETE. Rata-rata *response time* pada sistem unregistrasi adalah 726 ms, sedangkan rata-rata dari *response size* pada sistem unregistrasi adalah 377 bytes. *Response time* pada sistem unregistrasi dipengaruhi oleh koneksi dari stasiun ke server REST API melalui internet.

Analisis Pengujian Response time pada Server REST API

Berdasarkan pengujian yang telah dilakukan pada keseluruhan sistem, didapatkan kesimpulan terhadap telah dihimpun sebelumnya. Hasil pada pengujian ini dipengaruhi oleh beberapa faktor, salah satunya adalah koneksi internet dan besar data yang ditransmisikan. Tabel 5 menjelaskan hasil penghimpunan seluruh data uji pada bagian rata-rata *response time* dan *response size*.

Tabel 5. Analisis pengujian *response time* pada server REST API

No	Nama Sistem	*Response time* (ms)	*Response size* (byte)
1	Sistem Registrasi	2056	422
2	Sistem Pengiriman Data Sensor	486	470
3	Sistem Pembacaan Data Sensor	562	1254
4	Sistem Unregistrasi	726	377
Rata-rata		958	631

Berdasarkan tabel 5, terdapat empat buah rata-rata dari *response time* dan *response size* pada tiap sistem pada server REST API. Rata-rata *response time* pada keseluruhan sistem adalah 958 ms, sedangkan rata-rata dari *response size* pada keseluruhan sistem adalah 631 bytes. Dapat disimpulkan bahwa *response time* pada keseluruhan sistem kurang dari 1 detik, sehingga dapat dikatakan server REST API merespon request dengan cepat.

Pengujian *Performance* Layanan Artikel *Wordpress*

Pengujian *performance* layanan artikel *wordpress* ini menguji performa *wordpress* dalam memberikan layanan situs web sesuai standar *website* yang baik. Pengujian ini menggunakan aplikasi Lighthouse. Aplikasi ini digunakan karena dapat memberikan informasi tentang performa *website* secara akurat dan detail Aspek yang diuji meliputi *performance*, accessibility, best practices, dan SEO.

Tabel 6. Pengujian *perfomance* pada server *wordpress* menggunakan Lighthouse

No	Aspek yang diuji	Nilai (tampilan *mobile*)	Nilai (tampilan *desktop*)
1.	*Performance*	73	80
2.	Accessibility	88	83
3.	Best Practices	92	92
4.	SEO	93	92
Rata-rata		86,5	86,75

Berdasarkan tabel 6, didapatkan rata-rata skor pada tampilan *mobile* adalah 86,5, sedangkan pada tampilan desktop, rata-rata skor adalah 86,75. Hal ini didasari dari pengolahan gambar dan pembuatan web yang disesuaikan dengan standar aplikasi *Lighthouse*. Semakin besar skor pada aplikasi *Lighthouse*, maka semakin bagus performa dari layanan artikel *wordpress*. Selain itu pada kedua hasil pengujian tersebut juga menunjukkan bahwa server Layanan Artikel *Wordpress* telah mendukung *Progressive Web App* (PWA) yang memungkinkan layanan ini untuk berjalan secara *offline*.

Pengujian Pengiriman Data dengan *Eddystone URL*

Pengujian pengiriman data dengan *Eddystone URL* menguji transmisi data dari masing-masing stasiun menuju *web client*. Hal ini diperlukan untuk melihat seberapa jauh jarak

maksimal *Bluetooth* yang dapat ditoleransi dengan protokol *Eddystone URL*. Pengujian ini juga menjadi penentu dari keberhasilan *Physical Web* dalam mengintegrasikan sistem pengenalan desa wisata berbasis web dengan server REST API. Pengujian ini meliputi jarak pengujian, RSSI, *advertising interval*, URL terkirim, dan status pengiriman. Pengujian ini mengambil 10 sampel data tiap stasiun, kemudian menghitung persentase keberhasilan pengiriman data dari setiap sampel data tersebut.

Pengujian Pengiriman Data pada Stasiun Monitoring 1

Pengujian pengiriman data pada Stasiun Monitoring 1 dilakukan dengan cara menaruh stasiun pada sebuah tempat yang luas. Setelah itu mengamati parameter-parameter pada *Bluetooth* menggunakan aplikasi *BLE Scanner*. Hasil pengamatan tersebut dihimpun pada tabel 7.

Tabel 7. Pengujian pengiriman data pada stasiun monitoring 1

No	Jarak Pengujian (meter)	RSSI (dBm)	*Advertising interval* (ms)	Status Pengiriman
1	1	-37	182	berhasil
2	2	-43	162	berhasil
3	3	-42	166	berhasil
4	4	-40	168	berhasil
5	1	-43	729	berhasil
6	2	-31	189	berhasil
7	3	-39	157	berhasil
8	1	-51	135	berhasil
9	2	-43	470	berhasil
10	3	-63	351	berhasil

Berdasarkan tabel 7, pengujian pengiriman data pada Stasiun Monitoring 1 berhasil mengirim data dengan 10 kali berhasil dan 0 kali gagal, sehingga dapat disimpulkan bahwa pengujian pengiriman data tersebut memiliki persentase 100%. Pada pengujian ini nilai tertinggi dari RSSI adalah -31 dBm dengan jarak pengujian 2 meter, sedangkan nilai terendah dari RSSI adalah -63 dBm dengan jarak pengujian 3 meter. *Advertising interval* terbesar dari data pengujian ini adalah 470 ms, dimana nilai ini masih memenuhi standar *Physical Web* untuk berfungsi.

Pengujian Pengiriman Data pada Stasiun Monitoring 2

Pengujian pengiriman data pada Stasiun Monitoring 1 dilakukan dengan cara menaruh stasiun pada sebuah tempat yang luas. Setelah itu mengamati parameter-parameter pada *Bluetooth* menggunakan aplikasi *BLE Scanner*. Hasil pengamatan tersebut dihimpun pada tabel 8.

Tabel 8. Pengujian pengiriman data pada stasiun monitoring 2

No	Jarak Pengujian (meter)	RSSI (dBm)	*Advertising interval* (ms)	Status Pengiriman
1	1	-46	235	berhasil
2	2	-71	263	berhasil
3	3	-74	166	berhasil
4	4	-74	152	berhasil
5	1	-41	102	berhasil
6	2	-55	125	berhasil
7	3	-39	124	berhasil
8	1	-75	102	berhasil
9	2	-62	113	berhasil
10	3	-67	158	berhasil

Berdasarkan tabel 8, pengujian pengiriman data pada Stasiun Monitoring 2 berhasil mengirim data dengan 10 kali berhasil dan 0 kali gagal, sehingga dapat disimpulkan bahwa pengujian pengiriman data tersebut memiliki persentase 100%. Pada pengujian ini nilai tertinggi dari RSSI adalah -39 dBm dengan jarak pengujian 3 meter, sedangkan nilai terendah dari RSSI adalah -75 dBm dengan jarak pengujian 1 meter. *Advertising interval* terbesar dari data pengujian ini adalah 235 ms, dimana nilai ini masih memenuhi standar *Physical Web* untuk berfungsi.

Pengujian Pengiriman Data pada Stasiun Kontrol 1

Pengujian pengiriman data pada Stasiun Monitoring 1 dilakukan dengan cara menaruh stasiun pada sebuah tempat yang luas. Setelah itu mengamati parameter-parameter pada *Bluetooth* menggunakan aplikasi *BLE Scanner*. Hasil pengamatan tersebut dihimpun pada tabel 9.

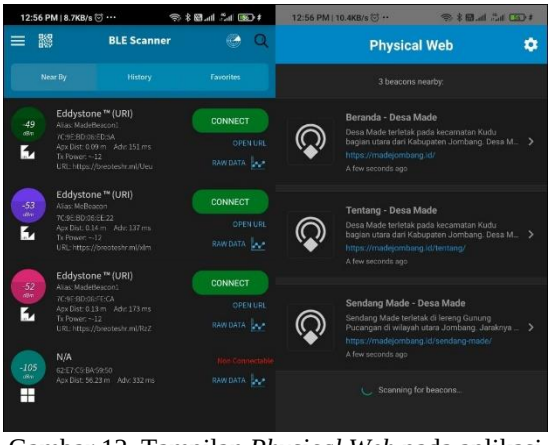
Tabel 9. Pengujian pengiriman data pada stasiun kontrol 1

No	Jarak Pengujian (meter)	RSSI (dBm)	*Advertising interval* (ms)	Status Pengiriman
1	1	-68	311	berhasil
2	2	-63	160	berhasil
3	3	-50	283	berhasil
4	4	-40	203	berhasil
5	1	-54	177	berhasil
6	2	-67	184	berhasil
7	3	-69	243	berhasil
8	1	-55	181	berhasil
9	2	-46	243	berhasil
10	3	-49	307	berhasil

Berdasarkan tabel 9, pengujian pengiriman data pada Stasiun Monitoring 1 berhasil mengirim data dengan 10 kali berhasil dan 0 kali gagal, sehingga dapat disimpulkan bahwa pengujian pengiriman data tersebut memiliki persentase 100%. Pada pengujian ini nilai tertinggi dari RSSI adalah -31 dBm dengan jarak pengujian 2 meter, sedangkan nilai terendah dari RSSI adalah -63 dBm dengan jarak pengujian 3 meter. *Advertising interval* terbesar dari data pengujian ini adalah 470 ms, dimana nilai ini masih memenuhi standar *Physical Web* untuk berfungsi.

Analisis Pengujian Pengiriman Data Menggunakan Eddystone URL

Berdasarkan hasil pengujian pengiriman data pada ketiga stasiun, didapatkan total tiga tabel dengan masing-masing 30 data uji. Pada stasiun monitoring 1, *advertising interval* terbesar adalah 470 ms dengan keberhasilan pengiriman data 100%, sedangkan pada stasiun monitoring 2, *advertising interval* terbesar adalah 235 ms dengan keberhasilan pengiriman data 100%. Kemudian pada stasiun kontrol 1, *advertising interval* terbesar adalah 311 ms dengan keberhasilan pengiriman data 100%. Didapatkan kesimpulan bahwa keberhasilan pengiriman data pada ketiga stasiun adalah 100% dengan rata-rata *advertising interval* tertinggi adalah 339 ms. Rata-rata *advertising interval* ini menunjukkan bahwa data berhasil ditransmisikan sesuai dengan standar *Physical Web* di mana waktu *advertising interval* direkomendasikan tidak lebih dari 700 ms, sehingga pada implentasi *Physical Web* dapat berjalan dengan lancar. Gambar 12 menunjukkan ketiga stasiun yang mengaktifkan *Physical Web*.



Gambar 12. Tampilan *Physical Web* pada aplikasi *mobile*

KESIMPULAN

Berdasarkan perancangan dan evaluasi *Physical Web* yang terintegrasi dengan REST API pada Sistem Pengenalan Desa Wisata, didapat kesimpulan berikut:

1. Pengujian *Physical Web* pada tiap stasiun melalui pengujian pengiriman data menunjukkan rata-rata *advertising interval* tertinggi sebesar 339 ms. Nilai ini telah memenuhi standar transmisi data dari *Physical Web* dimana *advertising interval* yang direkomendasikan kurang dari 700ms. Selain itu, pengujian pengiriman data menunjukkan keberhasilan pengiriman data sebesar 100% pada tiap stasiun.
2. Pengujian *performance* pada Layanan Artikel *Wordpress* menggunakan *Lighthouse* menunjukkan rata-rata skor pada tampilan *mobile* adalah 86.5, sedangkan pada tampilan *desktop*, rata-rata skor adalah 86.75.
3. Pengujian *response time* tiap sistem pada server REST API menunjukkan rata-rata *response time* pada keseluruhan sistem adalah 958 ms, sedangkan rata-rata dari *response size* pada keseluruhan sistem adalah 631 bytes.
4. Pengujian fungsionalitas server REST API menunjukkan bahwa server REST API dapat menerima, mengolah data, dan mengirim data sebagaimana mestinya. Sistem registrasi stasiun dan penyimpanan sensor sudah berjalan dengan baik. Hal ini juga membuktikan bahwa server REST API berhasil melakukan fungsi registrasi dan penyingkatan URL untuk mengaktifkan layanan *Physical Web*.

DAFTAR PUSTAKA

ADDIN Mendeley Bibliography CSL_BIBLIOGRAPHY Google. (2016). *Physical Web*. https://google.github.io/physical-web/

Google. (2017). *Eddystone*. https://github.com/google/eddystone

Hanani, A., & Hariyadi, M. A. (2020). Smart Home Berbasis IoT Menggunakan Suara Pada Google Assistant. *Jurnal Ilmiah Teknologi Informasi Asia*, *14*(1), 49. https://doi.org/10.32815/jitika.v14i1.456

Khasoggi, B. (2017). A Web of Things dengan RESTful protokol pada Smart City. *Prosiding Annual Research Seminar 2017*, *3*(1), 4–5.

Mahayadnya, G. A. P., Afkariansyah, G. R., Hartanto, F. F., Syahjaya, M. L., & Harianto, H. (2021). Pengembangan

Discoverability Pada Sistem Deteksi Banjir Kiriman Menggunakan Protokol *Eddystone URL* Berbasis Web of Things. *Antivirus : Jurnal Ilmiah Teknik Informatika*, *15*(2), 147–162. https://doi.org/10.35457/antivirus.v12i2.1611

Mubariz, A., Nur, D., Tungadi, E., & Utomo, M. N. Y. (2020). Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node . JS (Studi Kasus : Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang). *Seminar Nasional Teknik Elektro Dan Informatika (SNTEI)*, 72–77.

Pratama, F. G., & Kurnia, G. (2018). STRATEGI PENGEMBANGAN DESA WISATA BERBASIS MASYARAKAT (Studi Kasus: Desa Lebakmuncang, Kecamatan Ciwidey, Kabupaten Bandung). *Jurnal Ilmiah Mahasiswa AGROINFO GALUH*, *5*(1), 1014–1028.

Pratiarso, A., Mahendra, T. A., Yuliana, M., Kristalina, P., Astawa, I. G. P., & Arifin, A. (2018). Implementasi Sistem Notifikasi untuk Pengawasan Pasien Alzheimer Berbasis *Bluetooth* Low Energy (BLE). *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, *7*(4), 411–417. https://doi.org/10.22146/jnteti.v7i4.459

Rahmatillah, T. P., Insyan, O., Nurafifah, N., & Hirsan, F. P. (2019). Strategi Pengembangan Desa Wisata Berbasis Wisata Alam dan Budaya Sebagai Media Promosi Desa Sangiang. *Jurnal Planoearth*, *4*(2), 111. https://doi.org/10.31764/jpe.v4i2.970