

Implementasi *Middleware* Pada Evomo Dengan Metode *Web Service Restfull* Dan Pengujian *CI/CD*, *Coverage* Serta Simulasi Protokol Grafana

SYAFRIAL FACHRI PANE¹, AMRI YANUAR², ALIT FAJAR KURNIAWAN³, MAULYANDA⁴

^{1,2,3,4}Politeknik Pos Indonesia, Indonesia
Email : syafrial.fachri@poltekpos.ac.id

Received 27 September 2021 | *Revised* 21 Oktober 2021 | *Accepted* 21 Oktober 2021

ABSTRAK

Penelitian ini membantu dalam melakukan proses analisis data *monitoring* dalam bentuk grafik, memanfaatkan *CI/CD pipeline* dalam melakukan implementasi *CI/CD* dapat memberikan kenyamanan dalam melakukan *develop* dan mengurangi *bug*, dari hasil implementasi *CI/CD* didapatkan hasil *statements* 93,33%, *branch* 100%, *functions* 87,88%, *lines* 94,92%. Grafik data disiapkan oleh grafana dalam bentuk *script iframe* diterapkan pada *code* program sistem. Penelitian ini menggunakan metodologi penelitian yang dapat menyatakan bahwa sistem yang dibangun dapat berfungsi dengan baik. Jadi, penelitian ini mampu menjawab permasalahan yang terjadi pada sistem Evomo.

Kata kunci: *Evomo, middleware, Web Service RestFull, Antares, CI/CD*

ABSTRACT

This research helps in carrying out the monitoring data analysis process in graphical form, utilizing the CI/CD pipeline in implementing CI/CD can provide convenience in developing and reducing bugs. functions 87.88%, lines 94.92%, Graphic data prepared by grafana in the form of iframe script applied to the system program code. This study uses a research methodology that can state that the system built can function properly. So, this research is able to answer the problems that occur in the Evomo system.

Keywords: *Evomo, middleware, web service RestFull, Antares, CI/CD*

1. PENDAHULUAN

Dalam melakukan *monitoring* suatu objek sebelumnya menggunakan jasa manusia secara manual untuk mengelola atau *monitoring* lingkungan, maka pada saat ini sudah dikembangkan agar bisa mengontrol dan *monitoring* lingkungan secara otomatis dengan bantuan teknologi **(Ontowirjo, Fauziah YQ and Poekoel, Vecky C and Manembu, Pinrolinvic DK and Robot, & F, 2018)**. Evomo (*Environment Monitoring System*) merupakan sebuah sistem yang dibangun oleh perusahaan PT. Telekomunikasi Indonesia di bawah naungan tim Jeager, Evomo mampu memberikan sebuah informasi mengenai kondisi yang ingin dipantau berdasarkan indikator yang telah ditentukan baik berupa suhu, kelembaban, tekanan udara, pemakaian listrik, jumlah debit air, integrasi *cctv*, dll **(Loh, et al., 2018)**. Evomo menjadi solusi dalam melakukan *monitoring* proses bisnis yang sedang dijalani **(Ray & Pratim, 2018)**. Evomo memanfaatkan CI/CD *pipeline* dalam melakukan pengembangan *software* sekaligus mengurangi risiko *bug* dalam setiap tahapan pengembangan **(Putra & Ade, 2019)**. Pada penelitian ini lebih mengedepankan dalam pembangunan *middleware* **(AshifuddinMondal, Md and Rehena, & Zeenat, 2018)** pada sistem evomo, Evomo menggunakan teknologi Lora sebagai protokol komunikasi data dari alat ke sistem yang kemudian data akan dikirim ke *gateway* dan kemudian disimpan pada *platform* antares dalam bentuk *device ID* **(Thamrin, et al., 2020)**, Antares dihubungkan dengan alat melalui *Hierarchical URL* **(Alpharisy, Kevin Farid and Soim, Sopian and Hadi, & Irawan, 2020)**. *RestFul service get* melakukan akses pada *data/resource* pada REST server dimana masing-masing data tersebut akan dibedakan oleh sebuah global ID **(Rohman, Faisal Roufa and Soebroto, Arief Andy and Kharisma, & Putra, 2018)**. Penelitian ini memanfaatkan implementasi CI/CD untuk melakukan proses pengecekan *code* program dan proses deploy sistem evomo **(Rangnau, Thorsten and Buijtenen, Remco v and Fransen, Frank and Turkmen, & Fatih, 2020)**. Evomo menggunakan Grafana dalam proses menampilkan data grafik hasil dari deteksi suhu dan kelembaban **(Adnan, Muhammad and Rohmah, & Nur, 2020)**, dengan cara mengambil dengan *Non-hierarchical URL* yang telah disediakan oleh Antares dan kemudian digunakan untuk implementasi data grafik pada grafana **(Nurrahman, Ardian and Nazori, & AZ, 2020)**.

Masalah yang dihadapi pada penelitian ini yaitu proses dalam melakukan implementasi *middleware* untuk melakukan perpindahan data dari *platform* Antares ke sistem evomo. Data yang ditampilkan pada *platform* antares dalam bentuk *device ID* belum dapat diteruskan secara langsung ke sistem evomo untuk dapat ditampilkan ke *user*, diperlukan sebuah *service GET* data untuk dapat menampilkan data agar data dapat ditampilkan ke *user* **(Fauzan, Mohamad Nurkamal and Adiputri, & Chandiany, 2020)**, Proses implementasi CI/CD dilakukan pada saat sistem telah selesai di *develop* kemudian dilakukan pengujian untuk pengecekan *code* yang berguna untuk meminimalisir *bug* dan selanjutnya dilakukan proses *deploy*, kemudian menggunakan grafana dalam membuat data dalam bentuk grafik agar mudah dibaca **(Vassallo, et al., 2020)**. Untuk dapat menampilkan data hasil *monitoring* kepada setiap *user* maka penelitian ini melakukan implementasi *middleware* dengan menggunakan metode *web service RestFull* **(Kurniawan, Donny and Bhawiyuga, Siregar, & Reza, 2018)** yang dapat mengambil data dari *platform* antares dan kemudian menggunakan grafana dalam proses menampilkan grafik data pada sistem **(Widagdo, Karolus Thias and Bayu, Teguh Indra and Susetyo, & Alfa, 2019)**.

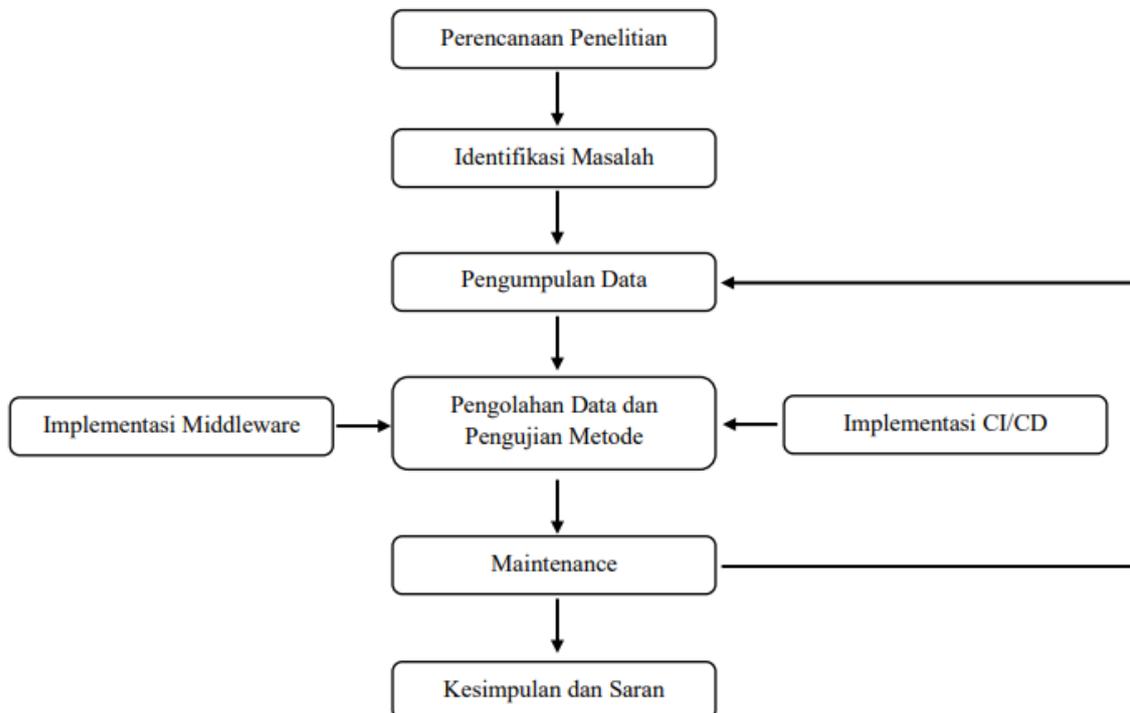
Untuk dapat menampilkan data hasil monitoring kepada setiap *user* maka penelitian ini melakukan implementasi *middleware* dengan menggunakan metode *web service RESTFUL* yang dapat mengambil data dari platform antares dan kemudian menggunakan Grafana dalam proses menampilkan grafik data pada sistem. *Web service* yang digunakan berbasis arsitektur REST kemudian dikenal dengan RESTFUL *web service*, Layanan web ini menggunakan CI/CD

pipeline dalam proses pengecekan *code* program yang digunakan untuk meminimalisir terjadinya bug dan proses melakukan *deploy* sistem.

Dari pembahasan tersebut dapat disimpulkan bahwa dengan menerapkan implementasi *middleware* yang menggunakan metode *web service RestFull* kemudian CI/CD pada proses *develop* untuk dapat meminimalisir terjadinya *bug* dan didapatkan hasil *statements* 93.33%, *branch* 100%, *functions* 87.88%, *lines* 94.92% dan proses *deploy* sistem, dan menggunakan grafana dalam pembuatan grafik untuk ditampilkan pada sistem evomo dan dapat membuat sistem berjalan dengan sebagaimana mestinya dan dapat memberikan kelengkapan untuk data hasil *monitoring* pada *Environment Monitoring System*.

2. METODE

Metodologi penelitian merupakan aturan kegiatan, dan prosedur yang digunakan oleh seorang peneliti. Metodologi juga merupakan sebuah analisa teoritis tentang suatu cara. Penelitian adalah suatu penyelidikan yang bersifat sistematis untuk meningkatkan sebuah pengetahuan, penelitian juga merupakan sebuah usaha yang sistematis dan terorganisasi untuk menyelidiki suatu masalah yang membutuhkan sebuah penyelesaian. Setiap orang mempunyai motivasi yang berbeda dalam melakukan penelitian, diantara-Nya dipengaruhi berdasarkan tujuan dan profesi dari setiap peneliti. Motivasi dan tujuan penelitian bersifat sama, yaitu bahwa penelitian adalah refleksi dari keinginan manusia yang selalu berusaha untuk dapat menciptakan dan mengetahui sesuatu yang baru. Keinginan Ketika ingin memperoleh dan mengembangkan pengetahuan merupakan kebutuhan dasar manusia yang menjadi motivasi untuk melakukan penelitian. Berikut ini adalah alur dari metodologi penelitian yang dilakukan di dalam penelitian ini, pada Gambar 1.



Gambar 1. Tahap – Tahap Yang Digunakan Pada Penelitian

2.1 Tahapan-Tahapan Diagram Alur Metodologi Penelitian

Tahapan Metode Penelitian yang dilakukan pada penelitian ini berdasarkan pada Diagram Alur Metodologi Penelitian di atas sebagai berikut :

a. Perencana Penelitian

Pada tahap ini merupakan tahapan awal yang akan dilakukan pada sebuah penelitian karena pada tahapan ini kita akan menentukan masalah apa yang akan kita jadikan penelitian dan menggunakan metode apakah penelitian ini dilakukan.

b. Identifikasi Masalah

Pada tahapan ini mengidentifikasi permasalahan yang terjadi pada sistem evomo dengan masukan data primer. Metode yang digunakan wawancara dan observasi, hasil keluaran yaitu rumusan masalah, ruang lingkup dan metodologi penelitian.

c. Mengumpulkan Data

Pada tahapan ini mengumpulkan data hasil *monitoring* mengenai suhu dan kelembapan untuk dilakukan pengujian dengan masukan data primer. Metode yang digunakan yaitu studi literatur, hasil keluaran yang didapatkan yaitu implementasi data hasil *monitoring* suhu dan kelembapan pada sistem evomo yang dapat ditampilkan kepada *user*.

d. Pengolahan Data dan Pengujian Metode

Pada tahapan ini Melakukan pengolahan dan pengujian data yang telah dikumpulkan dengan melakukan implementasi *middleware* menggunakan metode *Web Service RestFull* agar data yang disimpan pada *cloud* Antares dan dibuatkan grafik data menggunakan simulasi protokol Grafana agar dapat dengan mudah diimplementasikan di sistem Evomo kemudian melakukan implementasi CI/CD untuk dapat meminimalisir terjadinya *bug* dan melakukan *deploy*. Dan melakukan pengujian membuat grafik data pada Grafana dengan masukan data primer. Tahapan ini menggunakan metode *web service RestFull*, hasil keluaran yaitu data dapat ditampilkan pada sistem evomo.

e. Maintenance

Pada tahapan ini melakukan proses pengujian dengan menggunakan CI/CD *pipeline* dengan masukan data primer. Metode yang digunakan yaitu *web service RestFull*, dengan hasil keluaran mengurangi risiko kesalahan dalam melakukan implementasi *middleware*, simulasi protokol grafana dan proses *deploy* sistem.

f. Kesimpulan dan Saran

Tahapan kesimpulan dan saran merupakan akhir dari penelitian ini. Tahap ini memberikan kesimpulan dari hasil penelitian dan pengujian yang telah dilakukan pada penelitian ini.

3. HASIL DAN PEMBAHASAN

3.1 Mengumpulkan Data

Tahapan awal dalam proses penerapan metode *web service RestFull* yaitu mengumpulkan data dan mempersiapkan data yang akan digunakan berupa data suhu dan data kelembapan. Data harus diklasifikasi terlebih dahulu agar lebih sederhana dalam penggunaan data. Berikut data hasil klasifikasi pada tabel 1, setiap data dibedakan menjadi dua jenis yaitu data suhu dan data kelembapan, oleh karena itu setiap proses dalam mengumpulkan data maka harus dapat mengklasifikasikan data berdasarkan jenis data.

Tabel 1. Klasifikasi Data Suhu dan Kelembapan

No	Data
1	{ "suhu": 40, "kelembapan": "77" }
2	{ "suhu": 45, "kelembapan": "80" }
3	{ "suhu": 23, "kelembapan": "60" }
4	{ "suhu": 20, "kelembapan": "65" }

3.2 Pengolahan Data

Tahap pengolahan data merupakan tahapan yang dilakukan untuk mengolah data suhu dan kelembapan. Pengolahan data menggunakan metode pada *restfull*. Penelitian ini menggunakan 2 metode *restfull*, yaitu ada metode POST dan metode GET. Layanan web *RestFull* memanfaatkan kata kerja HTTP yang berat untuk menentukan operasi yang akan dilakukan pada sumber daya yang ditentukan. Berikut gambaran penggunaan metode POST dan GET pada proses pengolahan data untuk dapat digunakan.

a. Metode POST

Pada *method* POST melakukan pengiriman data langsung pada *action* untuk menampung data, data di input dari *form* yang telah disediakan, berikut *table 2* untuk penggunaan metode POST.

Tabel 2. Penggunaan Metode POST

Metode HTTP	POST
URL	https://xxx/~xxx/xxx-id/environment/simulasi
Operation	Mengirim Data
Operation-Type	-

b. Metode GET

Dalam penggunaan metode GET akan menampilkan data/nilai pada URL dan kemudian akan ditampung oleh *action*. Pada tabel 3 merupakan penggunaan metode GET.

Tabel 3. Penggunaan Metode GET

Metode HTTP	GET
URL	https://xxx/~xxx/xxx-id/environment/simulasi
Operation	Menerima Data
Operation-Type	Read Only

3.3 Penerapan Metode POST

Tahap pengolahan data merupakan tahapan yang dilakukan untuk mengolah data suhu dan kelembapan. Pengolahan data menggunakan metode pada *restfull*. Penelitian ini menggunakan 2 metode *restfull*, yaitu ada metode POST dan metode GET. Layanan web

RestFull memanfaatkan kata kerja HTTP yang berat untuk menentukan operasi yang akan dilakukan pada sumber daya yang ditentukan. Berikut penggunaan metode POST dan GET pada proses pengolahan data untuk dapat digunakan. Dan berikut hasil JSON yang didapatkan.

```
{
  "m2m:cin": {
    "rn": "cin_SccbgNR3To2GycFP",
    "ty": 4,
    "ri": "/xxx-cse/cin-SccbgNR3To2GycFP",
    "pi": "/xxx-cse/cnt-I0A2toDFRp-3KAut",
    "ct": "20210109T235643",
    "lt": "20210109T235643",
    "st": 0,
    "cnf": "text/plain:0",
    "cs": 29,
    "con": "{\"suhu\":40,\"kelembapan\": \"77\"}"
  }
}
```

3.4 Penerapan Metode GET

Pada Tahap ini penulis menerapkan metode GET dari *RestFull* terhadap penggunaan data suhu dan data kelembapan untuk digunakan menjadi data *monitoring* sistem *Environment Monitoring System*, dan menjadi penilaian juga apakah metode yang digunakan dapat memberikan hasil yang sesuai dengan yang diharapkan. Penerapan metode GET yang dilakukan menggunakan bantuan *postman* sebagai alat bantuan dalam pengujian permintaan API. Berikut penerapan metode *web service RestFull*. Untuk melakukan pengujian dalam penggunaan data pada antares maka kita lakukan terlebih dahulu pengujian pada *postman*, diharapkan membuka aplikasi *postman* dan membuat *workspace* baru dengan menggunakan *method* GET. Kemudian menambahkan alamat url dari *device* antares dan melakukan GET data. Maka akan didapatkan JSON seperti berikut.

```
{
  "m2m:cnt": {
    "rn": "simulasi",
    "ty": 3,
    "ri": "/xxx-cse/cnt-I0A2toDFRp-3KAut",
    "pi": "/xxx-cse/CAEr46qeAIDQv6vrIxu",
    "ct": "20210109T221755",
    "lt": "20210109T221755",
    "acpi": [ "/antares-cse/acp-35730207", "/xxx-cse/acp-438377616" ],
    "et": "20220109T221755",
    "st": 0,
    "mni": 30000000,
    "mbs": 10000,
    "mia": 0,
    "cni": 0,
    "cbs": 0,
    "ol": "/xxx-cse/xxx-id/environment/simulasi/ol",
    "la": "/xxx-cse/xxx-id/environment/simulasi/la" }
}
```

3.5 Penerapan CI/CD

Berikut merupakan proses dalam implementasi CI/CD pada penelitian ini. Implementasi CI/CD dilakukan menggunakan gitlab jadi proses yang dilakukan menjadi lebih mudah dan terstruktur.

Upload file project evomo ke repository gitlab

Kemudian masuk ke bagian *setting* -> CI/CD -> *runners* -> *expand* dan kemudian lihat *url* dan token akun kemudian jalankan dan sesuaikan *url* dan token anda.

```
sudo gitlab-runner register -n \
```

```
--url https://gitlab.com/  
--registration-token h_Sx1vRM1x5mWPsMsZhf  
--executor shell \  
--description "nodejs-app" \  
--tag-list "nodejs-app"
```

Kemudian lakukan *cloning*, dan buat *file .gitlab-ci.yml*

Isi dari *file .gitlab-ci.yml*

```
image: node:latest  
  
pre:  
  script: npm run test  
  artifacts:  
    when: always  
    paths:  
      - coverage  
    expire_in: 30 days  
  
cache:  
  paths:  
    - .cahce/  
    - node_modules/  
    - public/  
  
pages:  
  script:  
    - npm install  
    - ./node_modules/.bin/gatsby clean  
    - ./node_modules/.bin/gatsby build --prefix-paths  
  artifacts:  
    paths:  
      - public
```

Kemudian kita lakukan *push* ulang kembali ke repo yang sudah dibuatkan tadi, dengan cara masuk ke direktori *file*

```
Git add .  
Git commit -m "update"  
Git push origin master
```

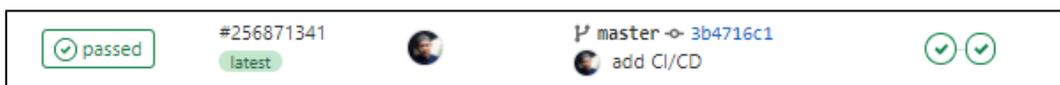
Kemudian masuk ke Gitlab dan masuk ke bagian *testing*, pada halaman *repository* pilih menu CI/CD dan masuk ke halaman *jobs*, dan lihat hasil *jobs success* dan mendapatkan hasil *statements* 93,33%, *branch* 100%, *functions* 87,88%, *lines* 94,92%.

Implementasi Middleware Pada Evomo Dengan Metode Web Service Restfull Dan Pengujian CI/CD, Coverage Serta Simulasi Protokol Grafana

File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s
	93.33	100	87.88	94.92	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	75	100	66.67	75	
	66.67	100	50	66.67	24,54
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	83.33	100	0	100	
	83.33	100	0	100	
	95	100	90.91	95	
	75	100	50	75	10
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	
	100	100	100	100	

Gambar 2. Hasil Perhitungan Code Coverage

Dan sukses dalam proses *deploy* dengan menggunakan Gitlab.



Gambar 3. Pipeline Deploy Dengan Gitlab

3.6 Penerapan Grafana

Dalam penggunaan grafik untuk menampilkan data maka kita menggunakan Grafana, Grafana merupakan sebuah *opensource* yang membaca sebuah data *matrices* untuk dibuat menjadi

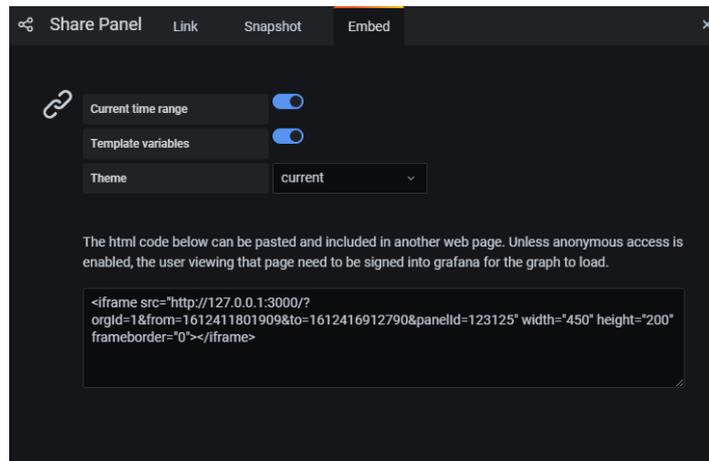
sebuah grafik atau sebuah data tertulis. Untuk melakukan analisis *monitoring* memang sangat tepat dalam menggunakan Grafana. Berikut merupakan simulasi penggunaan Grafana dalam implementasi grafik data yang tersimpan pada Antares.

Tahapan pertama yaitu kita menjalankan *software* Grafana dengan menjalankan *file* grafana-server.exe pada folder direktori Grafana. Kemudian melakukan login dan menambahkan panel, setelah menambahkan panel kemudian masuk ke edit panel dan melakukan setting pada panel. Kemudian kita membuatkan *link Non-hierarchical URL* dari *link cloud* penyimpanan data yang kita gunakan. Kemudian melakukan *apply* untuk melihat grafik, berikut hasil data grafik yang telah di lakukan oleh grafana pada gambar 4.



Gambar 3. Data Grafik Grafana

Dan berikut *script iframe* yang disediakan oleh grafana

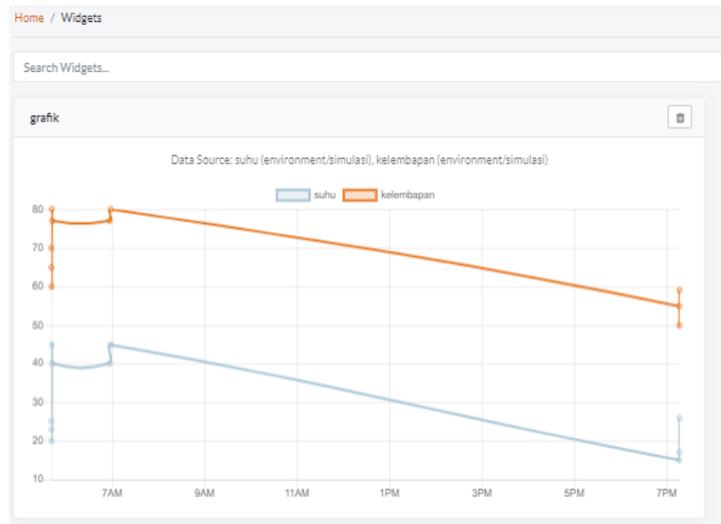


Gambar 4. Script Iframe

3.7 Hasil Penerapan Metode

Setelah melakukan pengolahan data dan penerapan metode *RestFull* dengan menggunakan perintah POST dan GET pada data suhu dan data kelembapan maka penulis memperoleh hasil bahwa data dapat ditampilkan pada sistem Evomo dan dapat dilihat oleh *user* dalam bentuk grafik dengan menggunakan simulasi protocol Grafana sebagai media pembuatan grafik data untuk dilakukan analisis oleh *user*.

Hasil yang didapatkan sesuai dengan harapan dari penelitian yang dilakukan. Berikut hasil dari data yang berhasil ditampilkan pada aplikasi pada gambar 6.



Gambar 5. Tampilan Data

4. KESIMPULAN

Berdasarkan pembahasan yang telah ditulis, proses analisis dan perancangan yang dilakukan, serta melakukan pengujian dengan metode *web service RestFull* untuk memberikan hasil penelitian yang sesuai dengan yang diinginkan. Penulis mendapatkan kesimpulan yaitu penerapan implementasi *middleware* menggunakan 2 metode dari *RestFull* yaitu POST dan GET yang dibuatkan *service* untuk proses pengolahan data, dengan tujuan dapat mengetahui proses dari penyederhanaan data dan proses perpindahan data ke sistem dengan menggunakan grafana dalam membuat grafik data dan diimplementasikan ke sistem Evomo. Dalam proses *develop* sistem juga berhasil menggunakan CI/CD sehingga sistem evomo dapat dijalankan dan meminimalisir *bug* dengan hasil dari perhitungan *code coverage* yaitu *statements* 93,33%, *branch* 100%, *functions* 87,88%, *lines* 94,92% dan berhasil melakukan proses *deploy*.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada program kerja sama magang BUMN tahun 2020 yaitu FHCI dari Politeknik Pos Indonesia terutama kepada PT. Telekomunikasi Indonesia yang telah membantu penulis dalam menyelesaikan penelitian ini.

DAFTAR RUJUKAN

- Adnan, Muhammad and Rohmah, & Nur, R. (2020). Pemanfaatan Dua Mikrokontroller Platform IoT Dalam Pengembangan Sistem Parkir. *Emitor: Jurnal Teknik Elektro*, 122--127.
- Alpharisy, Kevin Farid and Soim, Sopian and Hadi, & Irawan. (2020). Implementasi Sistem Monitoring Pemosisi Global Transportasi Darat Berbasis Esp32 Dengan Komunikasi Lora. *Jurnal Teknologi Technoscientia*, 68--75.
- AshifuddinMondal, Md and Rehena, & Zeenat. (2018). Iot based intelligent agriculture field monitoring system. In *2018 8th International Conference on Cloud Computing, Data Science |& Engineering (Confluence)* (pp. 625--629). IEEE.
- Fauzan, Mohamad Nurkamal and Adiputri, & Chandiany, L. (2020). *Tutorial Membuat Prototipe Prediksi Ketinggian Air (Pka) Untuk Pendeteksi Banjir Peringatan Dini Berbasis Iot*. Kreatif.
- Kurniawan, Donny and Bhawiyuga, Siregar, A. a., & R. A. (2018). Pengembangan Perangkat Middleware Web Of Things (WoT) Berbasis Arsitektur Publish Subscribe Menggunakan Protokol Rest HTTP. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN*, 964X.
- Loh, Kah Poh and McHugh, Colin and Mohile, Supriya G and Mustian, Karen and Flannery, Marie and Klepin, & Erika. (2018). Using information technology in the assessment and monitoring of geriatric oncology patients. *Current oncology reports*, 1--9.
- Nurrahman, Ardian and Nazori, & AZ. (2020). Sistem Kontrol Suhu Dengan Pid Dan Monitoring Daya Output Pada Panel Photovoltaic Portable Dengan Virtual Private Server. *MAESTRO*, 403--412.
- Ontowirjo, Fauziah YQ and Poekoel, Vecky C and Manembu, Pinrolinvic DK and Robot, & F, R. (2018). Implementasi Internet of Things Pada Sistem Monitoring Suhu dan Kelembaban Pada Ruang Pengerang Berbasis Web. *Jurnal Teknik Elektro Dan Komputer*, 331--338.
- Putra, & Ade, R. (2019). Analisa Implementasi Arsitektur Microservices Berbasis Kontainer Pada Komunitas Pengembang Perangkat Lunak Sumber Terbuka (Opendaylight Devops Community). *JUST IT: Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*, 150--162.
- Rangnau, Thorsten and Buijtenen, Remco v and Fransen, Frank and Turkmen, & Fatih. (2020). Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing

- Tools in CI/CD Pipelines. In *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)* (pp. 145--154). IEEE.
- Ray, & Pratim, P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, 291--319.
- Rohman, Faisal Roufa and Soebroto, Arief Andy and Kharisma, & Putra, A. (2018). Pengembangan Perangkat Lunak Aplikasi Monitoring Klimatologi Menggunakan Metode RESTful Web service Berbasis Android (Studi Kasus: Stasiun Klimatologi Karangploso Malang)}. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN*, 964X.
- Thamrin, Raihan Zaky and Samijayani, Octarina Nur and Rahmatia, Suci and Adrianto, Daniel and Enriko, & Agung, I. K. (2020). Implementation of LoRa End-Device in Sensor Network System for Indoor Application. In *2020 IEEE International Conference on Communication, Networks and Satellite (Comnetsat)* (pp. 208--212). IEEE.
- Vassallo, Carmine and Proksch, Sebastian and Jancso, Anna and Gall, Harald C and Di Penta, & Massimiliano. (2020). *Configuration smells in continuous delivery pipelines: a linter and a six-month study on GitLab*. Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.
- Widagdo, Karolus Thias and Bayu, Teguh Indra and Susetyo, & Alfa, Y. (2019). Pemodelan Sistem Monitoring Sensor Curah Hujan Menggunakan Grafana. *Indonesian Journal of Computing and Modeling*, 1--8.