Implementasi *Framework* Flask sebagai API Dalam Pengembangan *Website* Prediksi Kebakaran Hutan dan Lahan di Indonesia

1st Giovanni Nathaniel
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
gionathan@student.telkomuniversity.ac.id

2nd Casie Setianingsih
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

3rd Meta Kallista
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
metakallista@telkomuniversity.ac.id

***Abstrak*—Kebakaran hutan dan lahan di Indonesia merupakan salah satu masalah yang berdampak luas kepada berbagai aspek seperti kesehatan, lingkungan, ekonomi, dan sosial. Asap yang dihasilkan dari kebakaran hutan dapat menyebabkan penyakit pernapasan, menganggu ekosistem, dan hilangnya keanekaragaman hayati. Kebakaran hutan dan lahan juga dapat menyebabkan kerugian dari sisi ekonomi seperti peningkatan biaya pemadaman, merusak lahan pertanian. Oleh karena itu, dibutuhkan solusi pencegahan untuk meminimalisir risiko dari kebakaran hutan dan lahan di Indonesia. Solusi yang ditawarkan adalah dengan melakukan proses prediksi kebakaran hutan dengan memanfaatkan machine learning, Fire Weather Index (FWI), dan Geographical Information System (GIS). Dalam komunikasi antar sistem dibutuhkan API (Application Programming Interface) agar data dapat terus bergerak sesuai dengan kebutuhan sistem, terutama antara *frontend* dan *backend*. Dalam aplikasi ini API yang dipakai adalah flask yang memiliki basis bahasa python.**

***Kata kunci*— *backend*, flask, kebakaran, komunikasi**

I. PENDAHULUAN

Indonesia merupakan salah satu negara yang memiliki hutan terbanyak didunia, bahkan disebut sebagai paru-paru dunia dikarenakan hutan-hutan di Indonesia turut menyumbangkan oksigen untuk keberlangsungan kehidupan bagi makhluk hidup didunia. Namun, hutan yang ada di Indonesia semakin menurun tiap tahunnya diakibatkan oleh kejadian kebakaran hutan. Kebakaran hutan telah menjadi ancaman besar di seluruh dunia, menyebabkan banyak kerugian pada masyarakat dan ekosistem hutan [1]. Kebakaran hutan dan lahan disebabkan oleh dua faktor utama, yaitu manusia dan alam.

Faktor oleh manusia biasanya bertujuan untuk alih fungsi lahan untuk perkebunan maupun industri, sedangkan untuk faktor alam biasanya disebabkan oleh perubahan iklim yang ekstrim. Kebakaran hutan telah mengakibatkan kerugian dari berbagai aspek seperti kesehatan, ekonomi, dan lingkungan. Berdasarkan data dari Kementerian Lingkungan Hidup dan Kehutanan, tercatat luas kebakaran hutan dan lahan di Indonesia sepanjang Januari hingga Oktober 2023 yakni sekitar 994.313 hektare [2]. Selama periode antara 2019 dan Oktober 2023, kebakaran hutan dan lahan di Indonesia totalnya sudah mencapai 3.504.274 hektare. Kasus kebakaran hutan merupakan bencana tahunan yang sering terjadi di Indonesia. Oleh karena itu, dibutuhkan tindakan pencegahan kebakaran hutan agar dapat meminimalisir dampak yang ada.

Ada beberapa upaya yang telah dilakukan oleh pemerintah untuk menangani masalah kebakaran hutan dan lahan di Indonesai. Upaya tersebut terbagi menjadi tiga, yaitu pencegahan, pemadaman, dan penanganan pasca kebakaran. Tindakan pencegahan kebakaran hutan dan lahan harus dilakukan secara dini dan dibutuhkan dalam meminimalisir dampak yang akan dihasilkan ketika kebakaran terjadi [3].

Upaya pencegahan yang dapat dilakukan dalam meminimalisir risiko kebakaran hutan adalah dengan melakukan prediksi kebakaran secara akurat. Setelah itu, dilakukan perhitungan Fire Weather Index (FWI) yang dapat menilai potensi kebakaran hutan berdasarkan hasil prediksi dari keempat parameter yaitu suhu, kelembaban, curah hujan, dan kecepatan angin. FWI adalah komponen dari sistem peringatan kebakaran hutan yang digunakan untuk mengukur resiko kebakaran berdasarkan kondisi cuaca.

Sistem yang dikembangkan dalam memprediksi kebakaran hutan akan memanfaatkan algoritma machine learning untuk melakukan prediksi berbagai parameter dan pengembangan Geographical Information System (GIS). Untuk memberi segala data yang dibutuhkan untuk melakukan prediksi, menampilkan, serta menghitung nilai nilai FWI akan dibutuhkan perantara untuk menghubungkan antara *frontend* dan *backend*, yaitu berupa API (Application Programming Interface).

A. KAJIAN TEORI

Dalam pengembangan *website* prediksi kebakaran hutan dan lahan di Indonesia memanfaatkan banyak hal agar smeua sistem dapat berjalan dengan baik, terutama API (Application Programming Interface) yang berfungsi sebagai penghubung bagi data dari *backend* menuju *frontend.* Selain untuk menjadi penghubung API juga dibutuhkan untuk menghitung FWI (Fire Weathering Index)

sehingga membutuhkan sistem yang terpisah. Dengan menggunakan API sistem juga dapat berjalan lebih mulus dikarenakan tidak membebani sistem yang lain dalam proses perjalanan data, sehingga dengan pemanfaatan *library* dapat membuat sistem yang dapat diakses oleh sistem *software* lain [4].

a. Flask Framework

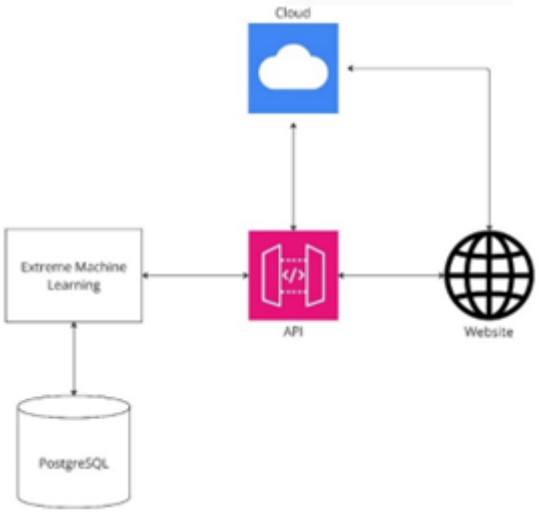
Flask Framework adalah suatu *framework* yang digunakan untuk membantu pengembangan web yang berbasis bahasa python [5]. Flask *framework* ini dapat digunakan sebagai *frontend* dan *backend* bagi sebuah *website*. Untuk pengembangan *website* prediksi kebakaran hutan dan lahan di Indonesia hanya menggunakan *backend* dari Flask, dikarenakan untuk memasukkan model *machine learning* berbasis bahasa python akan lebih mudah untuk diakses dengan *backend* yang juga berbasis bahasa python.

b. Fire Weather Index

Fire Weather Index atau disingkat menjadi FWI merupakan suatu sistem yang digunakan untuk mengukur potensi kebakaran hutan [6]. Ada beberapa parameter alam yang dibutuhkan dalam pengukuran index ini, yaitu: Suhu(°C), Kelembapan(%), Kecepatan angin(m/s), dan Curah hujan (mm). Dari keempat parameter ini akan dilakukan perhitungan menjadi FFMC, DC, DMC, ISI, dan BUI, hingga dari nilai ISI dan BUI menjadi nilai akhir yaitu nilai FWI.

B. METODE

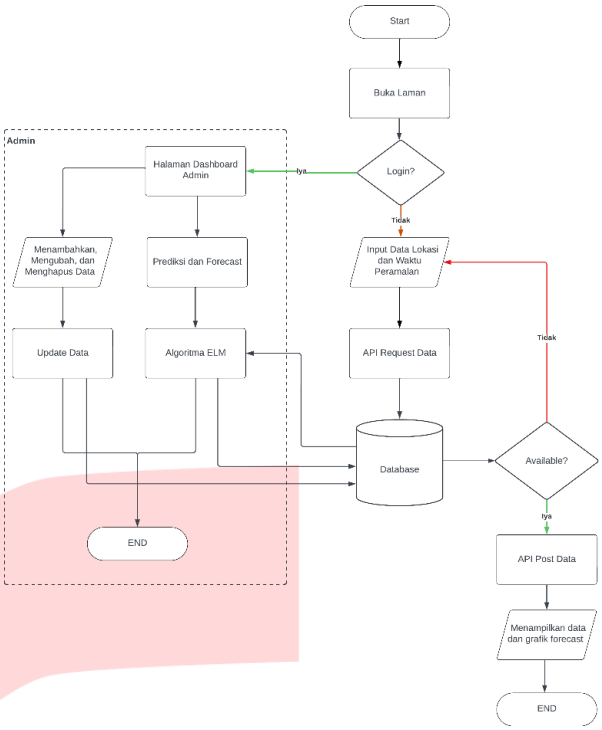
Berikut merupakan *flowchart* sistem dan arsitektur pada *website* prediksi kebakaran hutan dan lahan di Indonesia:



Gambar 1 Arsitektur Sistem

Di dalam sistem *website* prediksi kebakaran hutan dan lahan di Indonesia, Flask API difokuskan untuk menghubungkan antara *frontend* dan machine learning serta *database*.

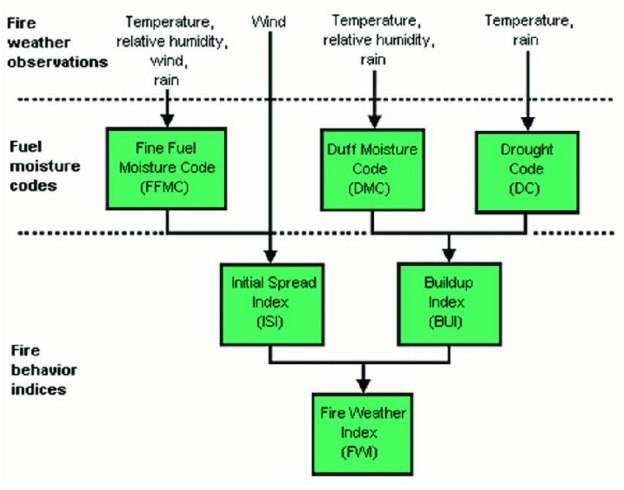
Berikut merupakan gambaran rancangan *flowchart* sistem dari *website* prediksi kebakaran hutan:



Gambar 2 Flowchart Sistem Website

Pada gambar diatas terlihat bahwa API cukup berperan penting dalam sistem bagi *user* dan *admin*. Hampir semua fitur untuk menampilkan data melibatkan *database* sehingga API dibutuhkan untuk mengambil data dari *database*, proses prediksi dan *forecast*, dan untuk perhitungan nilai FWI dari hasil prediksi dan *forecast.* Perhitungan FWI akan dibuat dalam kode berbahasa python hingga pada API hanya perlu memasukkan nilai parameter yang dibutuhkan serta nilai index di hari sebelumnya.

Berikut merupakan gambaran rancangan flow API yang akan diaplikasikan menggunakan flask *framework*:



Gambar 3 Proses perhitungan FWI

Pada Flowchart diatas menunjukkan proses perhitungan dari 4 parameter, yaitu: Suhu, Kelembapan, Kecepatan angin, dan Curah hujan hingga menjadi nilai FWI sebagai index potensi kebakaran hutan. Yang berawal dari nilai FFMC (Fine Fuel Moisture Code) yang terpengaruh oleh semua parameter, DMC (Duff Moisture Code) yang terpengaruh

suhu, kelembapan, serta curah hujan, dan DC yang terpengaruh suhu dan curah hujan. FFMC dilakukan perhitungan dengan menggunakan nilai kecepatan angin sehingga menjadi nilai ISI (Initial Spread Index), selain itu untuk DMC dan DC juga dilakukan perhitungan menjadi nilai BUI (BuildUp Index). Terakhir dari nilai ISI dan BUI dilakukan perhitungan menjadi nilai akhir FWI (Fire Weather Index).

II. HASIL DAN PEMBAHASAN

a. Hasil implementasai

Dikarenakan API perlu untuk terhubung pada *database*, dengan memanfaatkan *library* psycopg2 yang dikhususkan untuk koneksi pada postgreSQL.

```
connection = psycopg2.connect(user="ffp-indonesia",
                              password="*********",
                              host="34.***.***.221",
                              port="5432",
                              database="ffp-indonesia",
                              cursor_factory=RealDictCursor)
```

Gambar 4 Konfigurasi Databasae

Pada konfigurasi *database* diatas terdapat beberapa variabel yang harus disesuaikan dengan database yang sudah di *deploy*, yaitu: username, password, host ip, port, database name. Untuk koneksi *database* tersebut dimasukkan pada variabel "connection" untuk memberi perintah pada koneksi *databse* dapat dilakukan dengan memanggil variabel "connection".

Untuk dapat melaksanakan prediksi dan *forecast* diperlukan untuk *load* pada model machine learning

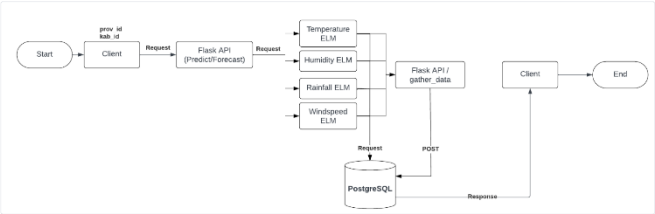
```
class CustomUnpickler(pickle.Unpickler):
    def find_class(self, module, name):
        if module == "__main__":
            module = "elmz"
        return super().find_class(module, name)


with open('/home/forestfirepredictionelm/ffp-api/modelV2.pkl', 'rb') as file:
    model = CustomUnpickler(file).load()
```

Gambar 5 Load Model Machine Learning

dengan memanfaatkan *library* pickle untuk membuat class CustomUnpickler untuk dapat mengakses class yang ada di dalam model bernama 'modelV2.pkl' yang disimpan pada /home/forestfirepredictionelm/ffp-api/. Dengan begitu, API dapat memanggil variabel "model" untuk melakukan *load* pada model machine learning.

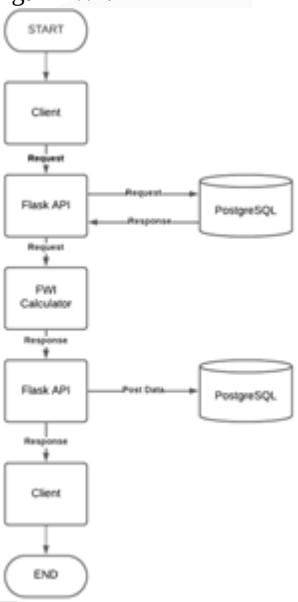
Berikut adalah flow dari proses eksekusi prediksi serta forecast:



Gambar 6 Flow API Prediksi dan Forecast

Pada gambar flow diatas dimulai dengan *frontend* yang melakukan *resquest* dengan cara mengakses *route* untuk prediksi atau *forecast*, serta memberi data berupa "prov_id" dan "kab_id". Dengan begitu pada *route* prediksi atau *forecast* utama melakukan *request* pada *route* tiap parameter untuk melaksanakan prediksi atau *forecast.* Contoh nama *route: "*/predict/humidity", dengan mengakses *route* tersebut berarti API akan melakukan proses prediksi pada parameter *humidity* (kelembapan). Sebelum melakukan prediksi diperlukan untuk mengambil data parameter pada *database* dengan memanfaatkan variabel "connection" dan juga perintah query, data yang diambil adalah data pada "prov_id" dan "kab_id" yang telah diberikan dari *frontend*, kemudian melakukan *data splitting* untuk menentukan data train dan data *test.* Pada tiap *route* parameter melakukan load pada machine learning dengan memanfaatkan variabel "model". Setelah proses selesai tiap *route* parameter akan mengirimkan data hasil proses kembali ke *route* induk, kemudian *route* induk akan mengurutkan data sesuai dengan tanggal. Setelah data diurutkan, data akan dikirim ke *database* yang khusus untuk menampung data-data *predict* dan *forecast* serta dikirim kembali dalam bentuk JSON.

Agar GIS dapat menampilkan warna sesuai dengan tingkat potensi kebakaran hutan, diperlukan juga perhitungan FWI pada API. Berikut adalah flow perhitungan FWI:



Gambar 7 Flow Perhitungan FWI

Disaat *frontend* melakukan *request* pada *route* perhitungan FWI, yaitu: /fetch_data_fwi, maka API akan melakukan akses pada database menggunakan variabel "connection" dan mengambil data semua parameter pada semua kabupaten yang tersedia. Kemudian, pada tiap kabupaten akan dilakukan perhitungan FWI menggunakan *file* python yang berbeda bernama "FWI Calculator". Untuk perhitungan FWI terdapat beberapa nilai awal yang telah ditetapkan oleh "Canadian Forest Fire Weather Index System"

```
ffmc0 = 85.0
dmc0 = 6.0
dc0 = 15.0
```

Gambar 8 Nilai Awal Perhitungan FWI

Berikut adalah beberapa keterangan dari nilai awal tersebut:

- ffmc0 = nilai awal FFMC, jika tidak ada nilai ffmc di hari sebelumnya. Bernilai 85;
- dmc0 = nilai awal DMC, jika tidak ada nilai ffmc di hari sebelumnya. Bernilai 6;
- dc0 = nilai awal DC, jika tidak ada nilai ffmc di hari sebelumnya. Bernilai 15.

Setelah selesai melakukan perhitungan semua index FWI pada tiap kabupaten, semua data hasil dikirim kembali ke *database* serta dikembalikan dalam bentuk JSON.

b. Pengujian sistem

Pengujian pada sistem dilakukan untuk memastikan sistem yang diimplementasikan dapat bekerja dengan dalam melakukan proses sesuai dengan fitur-fitur yang ada di *website* prediksi kebakaran hutan dan lahan di Indonesia.

Untuk pengujian sistem API dilakukan *Stress Testing*. *Stress Testing* adalah pengujian kinerja untuk mengevaluasi seberapa baik *website* dalam menangani beban kerja yang melebihi kapasitas. *Stress testing* dilakukan untuk memastikan bahwa sistem berjalan efektif dalam situasi kritis [7]. Pengujian ini dilakukan dengan tujuan unutk mengetahui batas maksimal *website* dan API dalam menjalankan *request.* Pada pengujian ini ada beberapa parameter utama, yaitu: jumlah *request* dan jumlah *virtual user*.

Dalam melakukan *stress testing* digunakan sebuah aplikasi bernama Postman yang berfungsi untuk mengirimkan *request* dan jumlah *virtual user* yang dapat diubah ubah. *Route* yang digunakan dalam pengujian adalah *route* dengan proses paling lama dan paling berat, yaitu: prediksi dan *forecast*. Pada *route* ini juga sudah termasuk untuk perhitungan FWI pada data hasi prediksi dan *forecast*. Untuk skenario yang dilakukan adalah sebagai berikut:

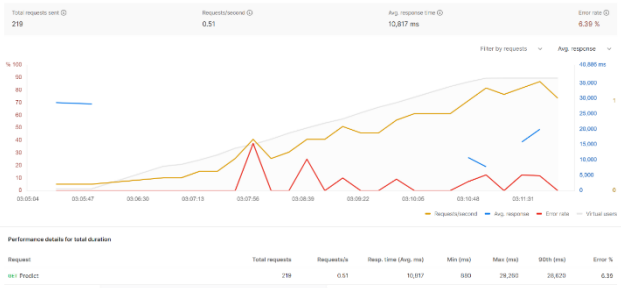
TABEL 1.

No	Skenario	Time	Method	Endpoint
1	60 *Virtual Users* (Ramp up)	7 Minutes	POST	https://forestfirepredictionidn.cloud/predict/train?selectedProvinsi=32&selectedKabupaten=3271
2	60 *Virtual Users* (Ramp up)	7 Minutes	POST	https://forestfirepredictionidn.cloud/forecast?selectedProvinsi=32&selectedKabupaten=3271

Pengujian dilakukan dengan metode POST pada *endpoint* /predict/train dan /forecast, masing-masing dengan parameter selectedProvinsi=32 dan selectedKabupaten=3271. Provinsi dengan kode 32 adalah Jawa Barat dan kota dengan kode 3271 adalah Kota Bogor. Setiap *endpoint* diuji dengan *Load Profile* mode Ramp up, yang memiliki arti jumlah *virtual users* akan terus naik selama waktu yang ditentukan sehingga dapat terlihat bagaiamana keadaan *website* pada tiap jumlah *virtual user*, sehingga ditentukan Ramp up 60 *virtual users* untuk mengukur kinerja API selama tujuh menit.

Hasil yang didapatkan dari pengujian *Stress Testing* adalah sebagai berikut:

1. Route Prediksi



Gambar 9 Grafik Stress Testing Route Prediksi

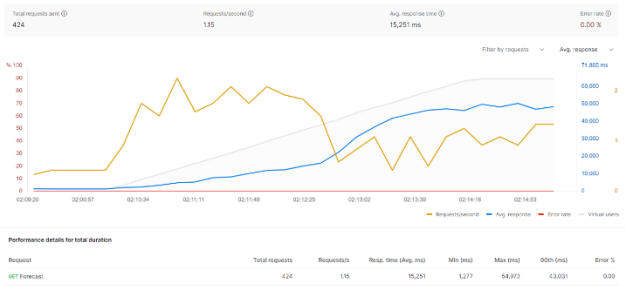
Berdasarkan grafik pengujian prediksi dengan mode Ramp up 60 *virtual users*, terdapat *error rate* yang tercatat selama pengujian sebesar 6.39 %. Selain itu, diperoleh *total requests sent* sebanyak 219 dengan *requests/second* sebesar 0.51. *Average response time* tercatat sebesar 10.817 s.

TABEL 2

Total Request Sent	Throughput	Response Time			Error Rate
		Average	Min	Max	
219	0.51 requests/s	10.8 s	0.8 s	29.2 s	6.39%

Berdasarkan hasil pengujian *stress testing* pada *route* prediksi, terlihat bahwa peningkatan jumlah *virtual users* berdampak pada performa sistem. Pengujian dengan mode Ramp up 60 *virtual user* terdapat *error rate* sebesar 6.39% dan *response time* rata-rata adalah 10.8 detik. Pada saat *virtual user* menyentuh di angka 24, *error* mulai terjadi dengan kode 500 Interval Server Error. Saat jumlah *virtual user* ditingkatkan hingga 60 *virtual user error* tetap terjadi dengan kode error yang sama. Hal ini menunjukkan bahwa terjadi penurunan kinerja seiring dengan peningkatan jumlah *virtual user*.

2. Route Forecast



Gambar 10 Grafik Stress Testing Route Prediksi

Berdasarkan grafik pengujian *forecast* dengan mode Ramp up 60 *virtual users*, tidak terdapat *error rate* yang tercatat selama pengujian. Selain itu, diperoleh *total requests sent* sebanyak 424 dengan *requests/second* sebesar 1.15. *Average response time* tercatat sebesar 15.251 s.

TABEL 3

Total Request Sent	Throughput	Response Time			Error Rate
		Average	Min	Max	
424	1.15 requests/s	15.2 s	1.2 s	54.9 s	0.0%

Berdasarkan hasil pengujian *stress testing* pada *route* prediksi, terlihat bahwa peningkatan jumlah *virtual users* berdampak sedikit pada performa sistem. Pengujian dengan mode Ramp up 60 *virtual user* tidak terdapat *error rate* dan *response time* rata-rata adalah 15.2 detik. Disaat *virtual user* menyentuh di angka 30, kecepatan *throughput* mulai mengalami penurunan hingga kecepatan *throughput* terendah terjadi disaat menyentuh 47 *virtual user*. Hal ini menunjukkan bahwa terjadi penurunan kinerja seiring dengan peningkatan jumlah *virtual user*, meskipun tidak terdapat *error rate* dan hanya mengalami penuruan *throughput*.

Data pengujian diatas dapat dianalisis berdasarkan tiap routenya. Berdasarkan hasil pengujian *stress testing* pada *route* prediksi menunjukkan penurunan performa yang cukup signifikan seiring dengan peningkatan jumlah *virtual users* (VU). Ketika dilakukan pengujian dengan mode Ramp up 60 VU terdapat *error rate* disaat menyentuh 30 VU dan *response time* rata-rata adalah 10,8 detik. Dari 30 VU hingga 60 VU selalu ditemukan *error* yang mengalami naik turun. Tingkat *error rate* ini disebabkan oleh beban komputasi yang sangat besar pada API prediksi, yang harus memproses sebanyak 2015 data, sehingga sistem tidak mampu menangani beban yang berat.

Route *forecast* menunjukkan kemampuan sistem dalam menangani peningkatan jumlah VU dengan baik tanpa *error rate* pada semua tingkat VU dari 1 VU hingga 60 VU. Pengujian dengan Ramp up 60 VU memiliki *response time* rata-rata 15,2 detik. Ketika jumlah VU menyentuh 24 nilai *throughput* mulai mengalami penurunan hingga titik terendah ada pada 47 VU. Hal ini karena data yang diproses oleh API *forecast* hanya mencakup prediksi untuk 7 hari ke depan, yang memerlukan waktu komputasi jauh lebih singkat dibandingkan dengan API prediksi. Oleh karena itu, perbedaan performa tidak terlalu signifikan bahkan tidak mengalami *error* sama sekali dan hanya menurunkan nilai *throughput*.

III. KESIMPULAN

Berdasarkan penelitian dan pengembangan yang telah dilakukan, dapat ditarik Kesimpulan. Sistem API (Application Programming Interface) yang dikembangkan menggunakan Flask Framework berhasil menyediakan solusi bagi fitur-fitur *website* Kebakaran Hutan dan Lahan di Indonesia dengan batas pengguna fitur prediksi dan *forecast* mencapai 60 pengguna. Hal ini menunjukkan bahwa dari sisi API *website* Kebakaran Hutan dan Lahan di Indonesia dapat digunakan oleh Masyarakat Indonesia.

REFERENSI

[1] F. Rasyid, “Permasalahan dan Dampak Kebakaran Hutan Fachmi Rasyid A. Pendahuluan,” Jurnal Lingkar Widyaiswara, vol. 1, no. 4, pp. 47–59, 2014.

[2] N. F. Hidayah, “Luas Kebakaran Hutan dan Lahan Indonesia Turun Pada 2022,” GoodStats. Accessed: Oct. 20, 2023. [Online]. Available: https://data.goodstats.id/statistic/Fitrinurhdyh/luas-kebakaran-hutan-dan-lahan-indonesia-turun-pada-2022-h5Qrs

[3] A. Ambarita, “Pencegahan Kebakaran Hutan dan Lahan Dalam Rangka Melindungi Pemukiman Masyarakat di Kabupaten Kotawaringin Barat Provinsi Kalimantan Tengah,” Jurnal Tatapamong, vol. 3, no. 1, pp. 56–78, Nov. 2021, doi: 10.33701/jurnaltatapamong.v3i1.1812.

[4] Matthias Biehl, API Architecture, vol. 2. API-University Press, 2015.

[5] “What is Flask Python.” Accessed: Aug. 09, 2024. [Online]. Available: https://pythonbasics.org/

[6] Natural Resources Canada, “Fire Weather Maps.” Accessed: Aug. 09, 2024. [Online]. Available: https://cwfis.cfs.nrcan.gc.ca/maps/fw?type=fwi

[7] I. Sontana, A. Rahmatulloh, and A. N. Rachman, “Application Programming Interface Google Picker Sebagai Penyimpanan Data Sistem Informasi Arsip Berbasis Cloud,” Jurnal Nasional Teknologi dan Sistem Informasi, vol. 5, no. 1, pp. 25–32, Apr. 2019, doi: 10.25077/TEKNOSI.v5i1.2019.25-32.