



## Implementasi Web Service untuk Aplikasi Pemantau Coronavirus Disease 2019 (COVID-19)

Faisal Al Isfahani<sup>1</sup>, Fuji Nugraha<sup>2</sup>, Rifki Mubarok<sup>3</sup>, Alam Rahmatulloh<sup>4</sup>

<sup>1,2,3,4</sup>Informatika, Fakultas Teknik, Univeristas Siliwangi, Tasikmalaya, Indonesia

<sup>1</sup>faisalfani56@gmail.com, <sup>2</sup>fujinugraha16@gmail.com, <sup>3</sup>rifkimubarok1410@gmail.com, <sup>4</sup>alam@unsil.ac.id

### INFORMASI ARTIKEL

#### Sejarah Artikel:

Diterima Redaksi: 18 Mei 2020

Revisi Akhir: 18 Mei 2020

Diterbitkan Online: 18 Mei 2020

### KATA KUNCI

Coronavirus,  
Pemantau,  
REST,  
Web Service

### KORESPONDENSI

Telepon: +62 85223519009

E-mail: alam@unsil.ac.id

### A B S T R A C T

The SARS-CoV-2 coronavirus outbreak, or commonly abbreviated as Covid-19, is currently almost paralyzing human life in various sectors. Compared to other coronaviruses such as SARS-Cov and MERS-Cov, Covid-19 spreads very quickly in multiple countries and regions. Also, the difficulty of information obtained accurately and in real-time for the community. The role of information technology is vital for the community in monitoring the coronavirus. This research utilizes web service technology to collect data and integrate services from various sources. The protocol and web service architecture used is representational state transfer (REST), with independent data exchange using the JSON format. Web service is made with node.js as well as android applications using an android studio. The results of this study have obtained a mobile-based android application that can present covid-19 data accurately and in real-time. Data recapitulation coronaviruses distribution from four service sources that have been integrated can display globally and locally in Indonesia.

## 1. PENDAHULUAN

Pandemi coronavirus 2019-2020 atau biasa dikenal dengan sebutan Covid-19 yaitu singkatan dari coronavirus disease 2019. Merupakan peristiwa penyebaran penyakit yang disebabkan coronavirus jenis baru yang diberi nama SARS-CoV-2 yang awalnya jenis virus baru ini bernama Novel Coronavirus 2019-nCov. Wabah yang disebabkan makhluk ukuran 80-150 nanometer ini mampu menyebabkan manusia dan perekonomian dunia nyaris berhenti bergerak, terhenti sampai waktu yang belum bisa ditentukan, dan kapan bisa kembali normal [1].

Wabah Covid-19 ditemukan pertama kali pada bulan Desember 2019 di Kota Wuhan, Hubei, Tiongkok China. Jika dibandingkan dengan coronavirus terkait sindrom pernafasan lain seperti SARS-Cov 2002/2003 dan MERS-Cov 2012-2014, coronavirus Covid-19 menyebar sangat cepat. SARS memerlukan waktu sekitar 4 bulan dan MERS memerlukan waktu yang lama sekitar dua setengah tahun untuk menginfeksi 1000 orang. Namun Covid-19 mampu mencapai angka itu hanya dalam 48 hari [2].

Pada tanggal 30 Januari 2020, Organisasi Kesehatan Dunia (WHO) telah menyatakan bahwa wabah coronavirus SARS-Cov-2 yang baru merupakan Public Health Emergency of International Concern (PHEIC) atau darurat

kesehatan masyarakat [3]. Dari tanggal 11 Maret hingga 16 Mei 2020 telah dilaporkan sebaran Covid-19 sebanyak 4.434.653 kasus dari seluruh dunia, bahwa lebih dari 216 negara dan wilayah yang mengakibatkan lebih dari 302.169 orang meninggal dunia, dan 781.109 orang lebih yang sembuh [4].

Sebaran wabah menyebar yang begitu cepat, dalam hal ini peranan teknologi informasi (IT) sangat diperlukan. Permasalahan utama adalah kurangnya informasi dan data sebaran Covid-19. Masyarakat membutuhkan pemantauan sebaran wabah covid-19 secara real-time. Saat ini teknologi yang paling banyak digunakan yaitu berbasis web dan *mobile*. Android berbasis mobile dapat menyajikan visualisasi yang lebih interaktif [5].

Diperlukan interface web untuk mencari data kasus covid-19 dari berbagai sumber yang terpercaya. Namun untuk dapat digunakan oleh masyarakat tentunya diperlukan interface yang lebih interaktif yaitu berbasis *mobile*. Diperlukan interkoneksi antara platform yang berbeda, untuk mengatasi permasalahan interoperabilitas antara aplikasi berbasis web dan *mobile* dapat menggunakan teknologi Web Service (WS) [6], [7].

Menurut beberapa penelitian, dengan teknologi *web service* mampu melakukan pertukaran data secara real-time [8], [9]. Penelitian ini akan dilakukan manipulasi dan

penggabungan data dari berbagai sumber pada *application programming interface (API)* yang berbentuk format JSON, kemudian data tersebut divisualisasikan pada aplikasi pemantau coronavirus secara real-time.

Hasil penelitian yang diharapkan yaitu adanya web service yang dapat dimanfaatkan oleh layanan lain serta aplikasi pemantauan coronavirus (Covid-19) yang dapat diakses dan digunakan oleh masyarakat. Sehingga informasi mengenai Covid-19 real-time.

**2. PENELITIAN TERKAIT**

Penelitian sebelumnya tentang penggunaan *web service* sudah banyak dilakukan, diantaranya oleh [10], [11] yang mengintegrasikan sistem akademik dan perpustakaan dengan menggunakan *Restful web service*. Walaupun sistem tersebut terpisah, namun dapat berkomunikasi dan bertukar data satu sama lainnya.

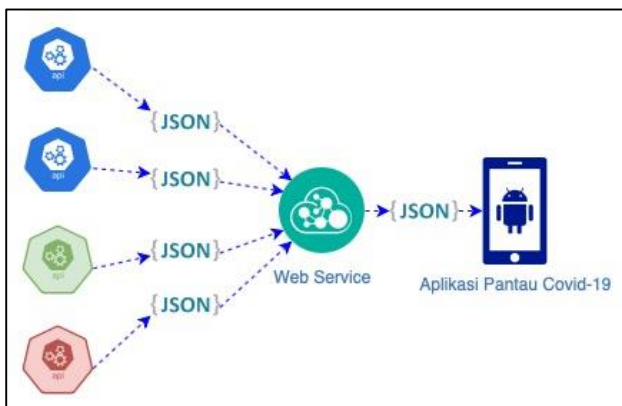
Penelitian lainnya telah dilakukan oleh [12] yang menggabungkan sistem pembayaran bank dan layanan akademik mahasiswa. Web service mampu mengatasi permasalahan interoperabilitas dengan tidak melihat jenis arsitektur, pemrograman, platform maupun sistem yang harus sama. Pertukaran data yang independen yaitu berformat JSON maupun XML, web service mampu mempertukarkan data tersebut dengan real-time.

Selain penelitian tersebut, masih banyak penelitian-penelitian yang telah dilakukan pada domain web service. Pada penelitian ini akan mencoba mengintegrasikan beberapa sumber data atau service yang berbeda mengenai covid-19, kemudian data-data tersebut disajikan dalam visualisasi berbasis mobile.

**3. METODE PENELITIAN**

Tahapan penelitian ini dibagi kedalam tiga tahap, yaitu perancangan dan arsitektur sistem, identifikasi kebutuhan sistem dan implementasi sistem.

**3.1. Perancangan dan Arsitektur Sistem**



Gambar 1. Arsitektur Sistem

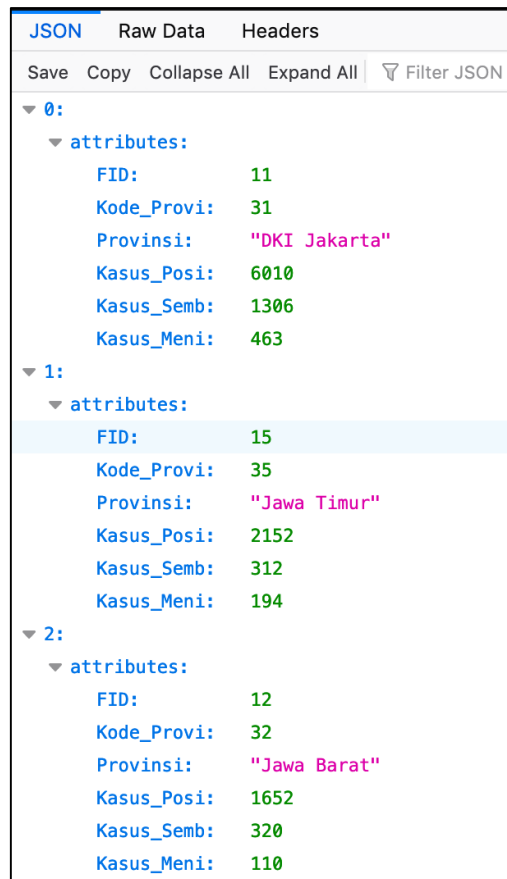
Arsitektur sistem yang akan dibuat dapat dilihat pada Gambar 1. Seluruh API tersebut diintegrasikan dalam satu web service dan disajikan dalam visualisasi aplikasi android. Masing-masing layanan yang diambil mempunyai fungsi dan peran sendiri. Integrasi dan pengambilan data coronavirus (Covid-19) dari empat sumber *application programming interface (API)* yang berbeda, dengan

masing-masing layanan tersebut secara rinci dapat dilihat pada Tabel 1.

TABEL I. MAPPING DAN INTEGRASI API

Layanan	Path	Integrasi
Kawal Corona	<a href="https://api.kawalcorona.com/indonesia/provinsi/">https://api.kawalcorona.com/indonesia/provinsi/</a>	Sebaran Data Wilayah Lokal Indonesia
Novel Covid API	<a href="https://disease.sh/v2/countries">https://disease.sh/v2/countries</a>	Sebaran Data Global Dunia
Javieraviles	<a href="https://coronavirus-19-api.herokuapp.com/all">https://coronavirus-19-api.herokuapp.com/all</a>	Rekap Data Global
Mathdroid	<a href="https://covid19.mathdro.id/api/countries">https://covid19.mathdro.id/api/countries</a>	Data Bendera Negara

Tabel 1 merupakan sumber data kasus coronavirus (Covid-19), dimulai dari layanan yang pertama yaitu kawal korona yang digunakan sebagai sebaran data wilayah Indonesia. RAW data tersebut dapat dilihat pada Gambar 2.



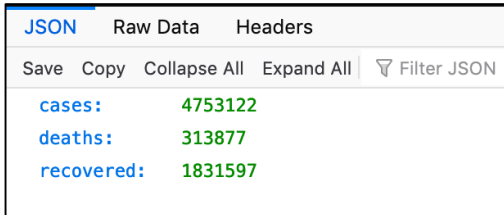
Gambar 2. JSON Sebaran Wilayah Lokal Indonesia

Layanan kedua yaitu Novel Covid API yang digunakan sebagai data sebaran Covid-19 dunia secara Global, dapat dilihat pada Gambar 3.

Layanan ketiga dari Javieraviles, sebuah akun github yang menyediakan rekap data kasus Covid-19 yang dapat dilihat pada Gambar 4. Data tersebut diambil per tanggal 17 Mei 2020 pukul 22.35 WIB dengan jumlah kasus 4.753.122, jumlah kematian 313.877 orang dan telah sembuh 1.831.597 orang.



Gambar 3. JSON Sebaran Wilayah Global



Gambar 4. JSON Rekap Data Global



Gambar 5. JSON Data Bendera Negara

Layanan terakhir bersumber dari Mathdroid yang digunakan sebagai visualisasi data bendera negara pada

aplikasi android, adapun raw data JSON dapat dilihat pada Gambar 5.

### 3.2. Identifikasi Kebutuhan Sistem

TABEL II. KEBUTUHAN PERANGKAT KERAS

Jenis	Minimum	Platform
Processor	Intel Celeron 1 GHz	Personal Computer
RAM	128 Mb	Personal Computer
Storage	128 Gb SSD	Personal Computer
Processor	200 MHz	Mobile (Android)
RAM	32 Mb	Mobile (Android)
Storage	32 Mb	Mobile (Android)

Identifikasi kebutuhan sistem untuk personal computer dalam pembuatan aplikasi pemantau coronavirus (Covid-19) dan aplikasi android dimulai dari perangkat keras (*hardware*) dan perangkat lunak (*software*), yang dapat dilihat pada Tabel 2 dan Tabel 3

TABEL III. KEBUTUHAN PERANGKAT LUNAK

Jenis	Minimum	Platform
OS	Windows 10	Personal Computer
Programming Language	Node.js	Personal Computer
Web Browser	Chrome/Firefox	Personal Computer
OS	Android 6 (Marshmallow)	Mobile (Android)

### 3.3. Implementasi Sistem

#### a. Pengkodean Web Service

Web service dibangun dengan Restfull menggunakan format JSON untuk pertukaran datanya. Adapun potongan kode sumber web service yang telah dibangun dapat dilihat sebagai berikut.

```
//data/dataSource.js
const axios = require("axios");
const fs = require("fs");

class DataSource {
  updateDataProvince() {

    axios.get("https://api.kawalcorona.com/indonesia/provinsi/").then((res) => {
      const date = new Date();
      const currentDate =
        date.toLocaleString("id-ID");
      const data = res.data;
      const datas = {
        data: data,
        updatedAt: currentDate,
      };
      const json = JSON.stringify(datas);
      fs.writeFile(__dirname + "/data.json",
        json, (err) =>
        err ? console.log(err) : ""
      );

      console.log("data was updated");
    });
  }

  updateDataWorld() {
    axios.get("https://coronavirus-19-api.herokuapp.com/all").then((res) => {
```

```

    const date = new Date();
    const current = date.toLocaleString("id-
ID");
    const datas = {
      data: res.data,
      updatedAt: current,
    };
    json = JSON.stringify(datas);
    fs.writeFile(__dirname + "/world.json",
json, (err) => {
      if (err) {
        console.err(err);
      }
    });
    console.log("data was updated");
  });
}

async updateDataCountries() {
  try {
    const date = new Date();
    const current =
date.toLocaleDateString("id-ID");
    const data = await axios
.get("https://coronavirus-19-
api.herokuapp.com/countries")
.then((res) => {
      return JSON.stringify({ data:
res.data, updatedAt: current });
    });
    fs.writeFile(__dirname +
"/data/countries.json", data, (err) =>
err ? console.log(err) : ""
);
    console.log("data was updated");
  } catch (err) {
    console.log(err);
  }
}
}

module.exports = DataSource;

```

Gambar 6. Pengambilan data API

Kode sumber `dataSource.js` pada Gambar 6 merupakan sebuah class yang mengambil data persebaran Covid-19, dimana terdapat tiga buah method untuk melakukan penyimpanan data pada file-file json. `updateDataProvince()` berfungsi untuk mengambil data sebaran coronavirus dari API lokal Indonesia (<https://api.kawalcorona.com/indonesia/provinsi/>) dan dimasukan pada file `data.json`. `updateDataWorld()` berfungsi untuk mengambil data persebaran Covid-19 secara global dari API (<https://coronavirus-19-api.herokuapp.com/all>) dan disimpan pada file `world.json`. Dan method `updateDataCountries()` berfungsi untuk mengambil data persebaran Covid-19 berdasarkan negara dari API (<https://disease.sh/v2/countries>) dan disimpan pada file `countries.json`. Class `DataSource` di export untuk digunakan pada main file `app.js`.

```

//app.js
const express = require("express");
const DataSource = require(__dirname +
"/data/dataSource");
const api = require(__dirname +
"/routers/api");
const cors = require("cors");

```

```

const port = process.env.PORT || 3000;

const dataSource = new DataSource();

//update data every 24 hour
setInterval(dataSource.updateDataProvince,
1000 * 60 * 60 * 24);
setInterval(dataSource.updateDataWorld, 1000 *
60 * 60 * 24);
setInterval(dataSource.updateDataCountries,
1000 * 60 * 60 * 24);

const app = express();
app.use(cors());
app.use(express.static(__dirname +
"/public"));
// use the api route
app.use("/api", api);
app.get("/", (req, res) => {
  res.sendFile(__dirname + "/index.html");
});

app.listen(port, () => console.log("server is
start on 3000"));

```

Gambar 7. Listening Request dari Client (app.js)

Gambar 7 merupakan kode sumber *listening request* dari *client*. File `dataSource.js` yang berisi *class DataSource* diinstansiasi pada constanta `dataSource` untuk melakukan pemanggilan method-method yang dimiliki class `DataSource`. Semua method `dataSource` dipanggil dan dilakukan *update* ke API utama dari Internet, setiap 24 jam sekali dengan pengaturan `setInterval()`, sehingga data sebaran Covid-19 akan selalu up-to-date sehari sekali. Pada saat `app.js` dijalankan maka aplikasi akan *listening* atau siap menerima *request* yang dilakukan oleh *client*, jika *client* merequest endpoint yang tidak terdapat pada *endpoint* yang disediakan maka *client* akan menerima response 404 not found.

Selesai semua layanan diintegrasikan, data akan disajikan dalam bentuk JSON. Hasil API yang telah dibuat pada web service ini dapat diakses pada url <https://arcane-chamber-95694.herokuapp.com/api/>. Web service yang dibuat dibagi kedalam lima endpoint.

*Endpoint* pertama <https://arcane-chamber-95694.herokuapp.com/api/> akan mengembalikan data global kasus Covid-19 dari mulai jumlah kasus, jumlah pasien meninggal, dan pasien yang sembuh.

*Endpoint* kedua <https://arcane-chamber-95694.herokuapp.com/api/countries> mengembalikan keseluruhan data persebaran Covid-19 dari seluruh dunia, lengkap dengan data yang terjadi per-hari ini.

*Endpoint* ketiga <https://arcane-chamber-95694.herokuapp.com/api/countries/:country> endpoint ini merensponse data berdasarkan negara, dengan mengganti `":country"` dengan negara yang akan di cek datanya.

*Endpoint* keempat <https://arcane-chamber-95694.herokuapp.com/api/provinsi> endpoint ini mengembalikan data persebaran Covid-19 spesifik untuk negara Indonesia, sama seperti endpoint-endpoint lainnya mengembalikan data jumlah kasus, jumlah pasien meninggal dan sembuh, dilengkapi dengan data kasus yang terjadi per-hari ini.

*Endpoint* kelima <https://arcane-chamber-95694.herokuapp.com/api/provinsi/:provinsi> endpoint ini

merespon data berdasarkan provinsi, dengan mengganti “:provinsi” dengan provinsi yang akan di cek datanya.

#### b. Pengkodean Aplikasi Android

Tahap selanjutnya adalah pembuatan kode sumber aplikasi *client* yaitu pada android. Adapun penjelasan potongan kode sumber yang telah dibangun dapat dilihat sebagai berikut.

```
package id.my.ter.covid19.api;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

// initial retrofit instance
public class RetrofitClientInstance {
    private static Retrofit retrofit;
    // end point api yang digunakan
    private static final String BASE_URL =
    "https://arcane-chamber-95694.herokuapp.com/api/";

    // pembuatan instance retrofit
    public static Retrofit
    getRetrofitInstance() {
        if (retrofit == null) {
            retrofit = new
            retrofit2.Retrofit.Builder()
                .baseUrl(BASE_URL)

            .addConverterFactory(GsonConverterFactory.crea
            te())

                .build();
        }
        return retrofit;
    }
}
```

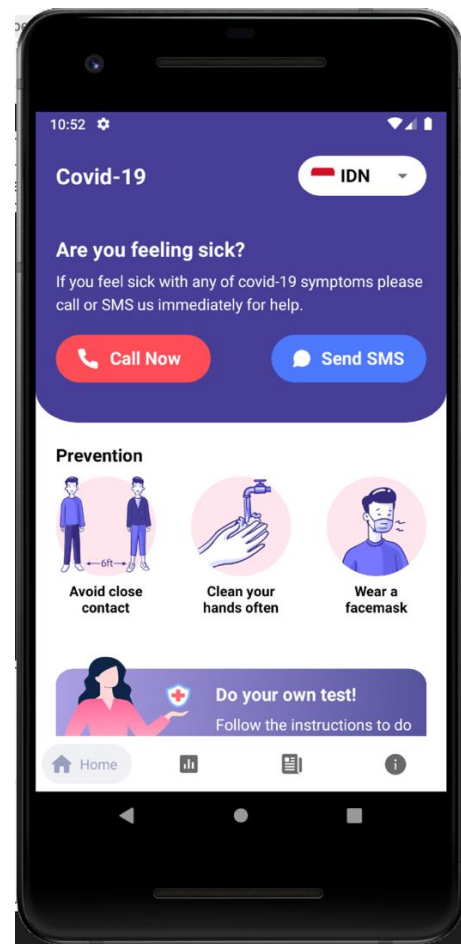
Gambar 8. Kode sumber request API

*Class* *RetrofitClientInstance* pada Gambar 8 merupakan Retrofit Instance yang berfungsi untuk melakukan request terhadap API, maka dari itu dibutuhkan Retrofit builder *Class* untuk Base URL dari sebuah layanan.

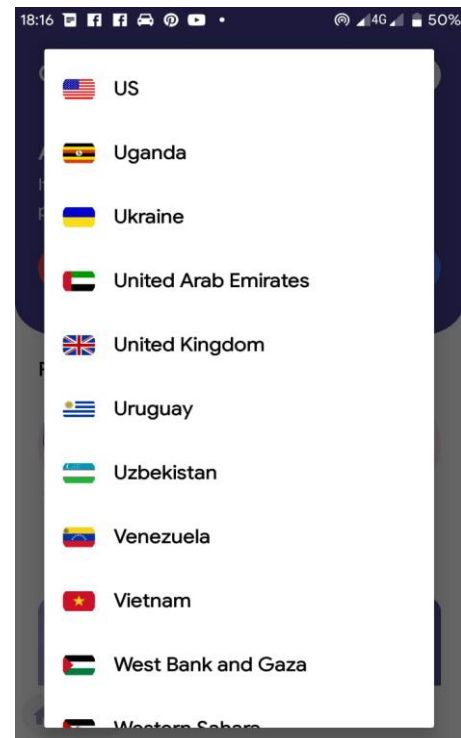
#### 4. HASIL DAN PEMBAHASAN

Setelah semua perancangan sampai implementasi sistem dilakukan, maka didapatkan hasil aplikasi yang telah dibangun seperti pada Gambar 9. Merupakan tampilan awal aplikasi pemantau coronavirus (Covid-19) yang terdiri dari empat menu, yaitu home, statistik, rekap dan informasi.

Tampilan utama aplikasi berada pada navigasi home, menunjukkan informasi mengenai contact emergency call Covid-19, dan beberapa informasi mengenai pencegahan penularan dengan ilustrasi. Di pojok kanan atas terdapat input selection untuk mengubah data persebaran Covid-19 berdasarkan negara yang dipilih.



Gambar 9. Tampilan awal aplikasi



Gambar 10. Pemilihan pemantauan sebaran berdasarkan negara

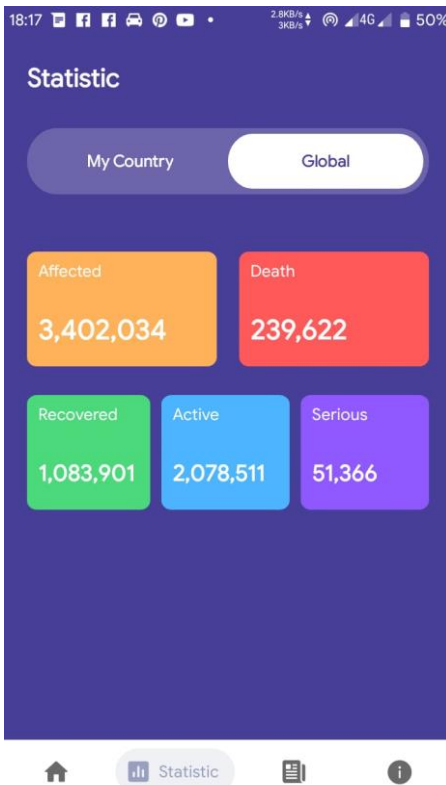
Sementara pada Gambar 10 merupakan menu pemilihan data sebaran coronavirus berdasarkan negara. Bendera yang muncul dalam visualisasi tersebut merupakan hasil dari layanan keempat. Ketika nama atau

bendera di klik maka akan menampilkan data statistik coronavirus pada negara tersebut.

Untuk menampilkan detail informasi mengenai jumlah data persebaran berdasarkan negara yang dipilih, menu tersebut terdapat pada navigasi statistic disamping kanan tombol navigasi home. Selain berdasarkan data persebaran berdasarkan negara terdapat tab Global yang menampilkan data persebaran Covid-19 secara global.



Gambar 11. Statistik lokal Indonesia

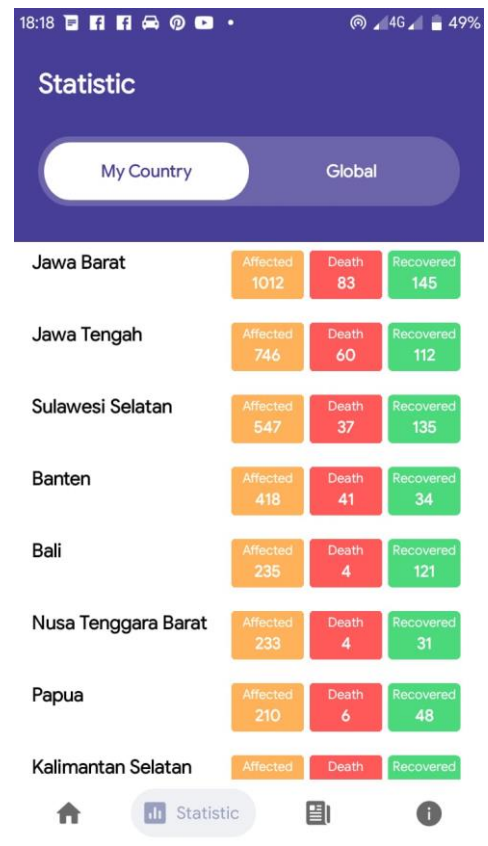


Gambar 12. Statistik Global

Jika menampilkan data berdasarkan negara Indonesia maka akan terdapat data persebaran lengkap berdasarkan provinsi yang ada di Indonesia. Adapun visualisasinya dapat dilihat pada Gambar 11 dan Gambar 12.



Gambar 13. Statistik Lokal Indonesia



Gambar 14. Data Sebaran Provinsi di Indonesia

Sementara untuk visualisasi sebaran data coronavirus wilayah Indonesia dan provinsinya dapat dilihat pada Gambar 13 dan Gambar 14. Pemantauan yang dapat dilakukan dengan melihat perkembangan dan sebaran mengenai jumlah kasus Covid-19, jumlah kematian dan jumlah yang telah sembuh akibat coronavirus (Covid-19).

## 5. KESIMPULAN

Penelitian ini telah menghasilkan dua produk yaitu web service Covid-19 yang dapat dimanfaatkan lagi bagi layanan lain. Aplikasi pemantauan coronavirus (Covid-19) dapat digunakan oleh masyarakat secara interaktif dan real-time. Integrasi dan pertukaran data yang terjadi pada sisi *backend* telah berhasil menggabungkan dari empat sumber layanan yang berbeda.

Perlu diskusi lebih mendalam mengenai lama waktu atau respon time, penggunaan memory serta jumlah data yang dipertukarkan dalam penelitian ini. Selain itu mekanisme pencarian data dari sumber yang kredibel perlu dilakukan secara otomatis, sehingga diperlukan mekanisme khusus dalam pengumpulan data tersebut.

## DAFTAR PUSTAKA

- [1] M. N. Kamel Boulos and E. M. Geraghty, "Geographical tracking and mapping of coronavirus disease COVID-19/severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) epidemic and associated events around the world: How 21st century GIS technologies are supporting the global fight against outbreaks and epidemics," *International Journal of Health Geographics*, vol. 19, no. 1, p. 8, Dec. 2020 [Online]. DOI: 10.1186/s12942-020-00202-8. [Accessed: 18-May-2020]
- [2] T. N. C. P. E. R. E. Team and T. N. C. P. E. R. E. Team, "The Epidemiological Characteristics of an Outbreak of 2019 Novel Coronavirus Diseases (COVID-19) — China, 2020," *China CDC Weekly*, 2020, Vol. 2, Issue 8, Pages: 113-122, vol. 2, no. 8, pp. 113–122, Feb. 2020. DOI: 10.46234/CCDCW2020.032
- [3] "Statement on the meeting of the International Health Regulations (2005) Emergency Committee regarding the outbreak of novel coronavirus 2019 (n-CoV) on 23 January 2020." [Online]. Available: [https://www.who.int/news-room/detail/23-01-2020-statement-on-the-meeting-of-the-international-health-regulations-\(2005\)-emergency-committee-regarding-the-outbreak-of-novel-coronavirus-\(2019-ncov\)](https://www.who.int/news-room/detail/23-01-2020-statement-on-the-meeting-of-the-international-health-regulations-(2005)-emergency-committee-regarding-the-outbreak-of-novel-coronavirus-(2019-ncov))
- [4] "Naming the coronavirus disease (COVID-19) and the virus that causes it." [Online]. Available: [https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-\(covid-2019\)-and-the-virus-that-causes-it](https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it)
- [5] H. Septian, E. W. Hidayat, and A. Rahmatulloh, "Aplikasi Pengenalan Bahasa Arab dan Inggris untuk Anak-Anak Berbasis Android," *Jurnal Online Informatika*, 2018. DOI: 10.15575/join.v2i2.100
- [6] A. Rahmatulloh, R. Gunawan, and I. Darmawan, "Web Services to Overcome Interoperability in Fingerprint-based Attendance System," in *Proceedings of the 2018 International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)*, 2019 [Online]. DOI: 10.2991/icoiese-18.2019.49
- [7] R. Gunawan and A. Rahmatulloh, "JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 5, no. 1, p. 74, 2019. DOI: 10.26418/jp.v5i1.27232
- [8] I. Darmawan, A. Rahmatulloh, H. Mubarak, R. Gunawan, and R. Syahrizani, "Real-time Communication Measurement on Web Services in the Fingerprint Machine," 2019, pp. 184–188.
- [9] A. Rahmatulloh, I. Darmawan, and R. Gunawan, "Performance Analysis of Data Transmission on WebSocket for Real-time Communication," *2019 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering*, pp. 1–5, Jul. 2019 [Online]. DOI: 10.1109/QIR.2019.8898135
- [10] R. Rizal and A. Rahmatulloh, "Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *Jurnal Ilmiah Informatika*, vol. 7, no. 01, p. 54, 2019. DOI: 10.33884/jif.v7i01.1004
- [11] F. Bimantoro, I. B. K. Widiartha, I. G. P. S. Wijaya, and A. Y. Husodo, "Integrasi Sistem Informasi Keuasan Belajar Mengajar Program Studi Teknik Informatika Dengan Sistem Informasi Akademik Unram Menggunakan Web Service," *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTika)*, 2019. DOI: 10.29303/jtika.v1i1.7
- [12] R. Gunawan and A. Rahmatulloh, "Implementasi Web Service pada Sistem Host-To-Host Pembayaran Biaya Akademik," *Setrum: Sistem Kendali-Tenaga-Elektronika-Telekomunikasi-Komputer*, vol. 7, no. 2, pp. 320–328, 2018. DOI: 10.36055/setrum.v7i2.4100

## BIODATA PENULIS



Faisal Al Isfahani

Mahasiswa program studi informatika di Universitas Siliwangi, dengan program keahlian Sistem dan Teknologi Informasi. Fokus pada Fullstack web development, pernah mengembangkan aplikasi tanggap bencana berbasis mobile (Merapi Minister).



Fuji Nugraha

Mahasiswa jurusan Informatika di Universitas Siliwangi dengan kelompok keahlian Sistem Teknologi Informasi, memiliki minat pada pemrograman, Software & Web Development, dan perancangan Sistem dan Teknologi Informasi.



Rifki Mubarak

Mahasiswa program studi informatika di Universitas Siliwangi. Fokus pada Web Programming dan Mobile Programming. Pernah mengembangkan Sistem Integrasi Guru Daerah Khusus berbasis web.



Alam Rahmatulloh

Dosen sekaligus peneliti pada bidang Teknologi Informasi. Fokus pada Microservices, Web Programming, IoT. Berpengalaman dalam pembuatan sistem informasi seperti smart campus, sistem akademik, dan lainnya.