

## PERANCANGAN SISTEM MONITORING PINTAR UNTUK MITIGASI PENULARAN COVID-19 DI AREA PADAT PUBLIK BERBASIS IOT

Depandi Enda<sup>1)</sup>, Muhammad Asep Subandri<sup>2)</sup>

<sup>1,2</sup>Teknik Informatika, Politeknik Negeri Bengkalis, Jl. Bathin Alam, Bengkalis, 28711  
E-mail: depandienda@polbeng.ac.id

### Abstract

Various ways can be done as an effort to mitigate the spread of COVID-19, one of which is by utilizing mobile or smartphone technology. Most applications built today use location-based technology (GPS) combined with communication protocols such as Bluetooth and WiFi. The challenge that arises from this system is the problem of wasteful use of battery resources and also the problem of violating user privacy information. This study aims to create a system design that provides solutions to the problems that have been described. The system that is built is an IoT-based smart monitoring system that is useful for mitigating COVID-19 transmission in crowded public areas. The system is designed to monitor the environment around crowded public areas by utilizing several sensors installed to read and record data on the surrounding environment such as temperature, humidity, noise level and location data in real time. Based on the results of data transmission testing, the percentage of packet lost is 0% and the accuracy of the recommended rules designed is 100%, this shows that the performance of the system is very well designed and can be implemented.

**Keywords:** monitoring system, IoT, COVID-19, mitigation, crowded public areas

### Abstrak

Berbagai cara bisa dilakukan sebagai upaya mitigasi penyebaran COVID-19, salah satunya adalah dengan cara memanfaatkan teknologi *mobile* atau *smartphone*. Kebanyakan aplikasi yang dibangun saat ini menggunakan teknologi berbasis lokasi (GPS) yang dikombinasikan dengan protokol komunikasi seperti Bluetooth dan WiFi. Tantangan yang muncul dari sistem ini adalah permasalahan penggunaan sumber daya baterai yang boros dan juga permasalahan pelanggaran informasi privasi pengguna. Penelitian ini bertujuan untuk membuat sebuah rancangan sistem monitoring pintar berbasis IoT yang berguna untuk mitigasi penularan COVID-19 di area padat publik. Sistem yang dirancang akan memonitor lingkungan sekitar area padat publik dengan memanfaatkan beberapa sensor yang terpasang untuk membaca dan merekam data lingkungan sekitar seperti suhu, kelembaban, level kebisingan dan data lokasi secara *realtime*. Berdasarkan hasil pengujian pengiriman data yang dilakukan diperoleh persentase *packet lost* sebesar 0% dan akurasi aturan rekomendasi yang dirancang sebesar 100%, ini menunjukkan bahwa kinerja sistem yang dirancang sangat baik dan dapat diimplementasikan.

**Kata Kunci:** sistem monitoring, IoT, COVID-19, mitigasi, area padat publik

## PENDAHULUAN

Berbagai cara bisa dilakukan sebagai langkah mitigasi penyebaran COVID-19 di area padat publik, salah satunya adalah dengan cara memanfaatkan teknologi aplikasi *mobile* pada perangkat *smartphone*. Aplikasi *mobile* yang dibangun pada umumnya digunakan untuk membantu dalam proses menemukan

atau penjelajahan orang yang mungkin memiliki riwayat kontak langsung maupun tidak langsung dengan individu yang terkonfirmasi positif virus (*contact tracing*).

Banyak aplikasi telah dikembangkan oleh pengembang, penggiat *mobile app*, perusahaan *start up* non-profit, bahkan akademisi juga turut andil, baik yang bergerak secara individu maupun bekerja sama dengan pemerintah dalam rangka membantu mengurangi dampak COVID-19. Kebanyakan aplikasi *mobile* yang dibangun adalah berbasis penjelajahan lokasi geografis (*maps*) yang dikombinasikan dengan teknologi *bluetooth*. Tantangan yang muncul dari sistem aplikasi *mobile* yang dibangun berbasis lokasi geografis dengan memanfaatkan teknologi *bluetooth* adalah permasalahan sumber daya baterai yang cepat habis dan juga permasalahan pelanggaran/pencurian informasi privasi pengguna. Hal ini dikarenakan dengan menghidupkan *bluetooth* dan layanan lokasi (GPS) secara terus menerus oleh aplikasi dapat menguras pemakaian baterai *smartphone*. Selain itu, pembagian data lokasi secara *realtime* oleh aplikasi juga dapat menimbulkan resiko adanya pencurian informasi pribadi oleh oknum yang tidak bertanggung jawab jika sistem yang dibangun tidak memiliki keamanan sistem yang baik.

Berikut adalah beberapa *framework* dan aplikasi penjelajahan kontak yang telah dikembangkan, antara lain *Framework* Google/Apple Privacy-Preserving Contact Tracing, kerangka kerja berbasis *bluetooth low energy* ini dirancang khusus oleh Google dan Apple sebagai solusi terhadap pandemi COVID-19 dengan memfokuskan pada privasi dan keamanan data pengguna ketika menjalankan layanan penjelajahan kontak di aplikasi (Google & Apple, 2020). Protokol aplikasi *Blue Trace*, merupakan protokol komunikasi *bluetooth* yang bersifat *open source* yang memfasilitasi penjelelahan kontak partisipan yang terinfeksi secara digital. Protokol aplikasi ini dikembangkan oleh otoritas Singapura sebagai layanan terhadap aplikasi TraceTogether dalam rangka memerangi wabah COVID-19 (Bay et al., 2020).

TraceTogether, sebuah aplikasi yang dirilis oleh otoritas Singapura yang memungkinkan penjelajahan kontak secara digital dengan menggunakan protokol komunikasi *BlueTrace* yang telah di kostumisasi. Adapun jangkauan kontak yang

dapat dilacak menggunakan aplikasi ini bisa mencapai 10 meter (33 kaki). (Government Technology Agency Singapore, 2020).

*Alipay Sign a Color Code App* adalah aplikasi buatan Alibaba Group, aplikasi ini memberikan informasi kepada pengguna tentang status suatu tempat/lokasi dengan menggunakan tanda hijau, kuning dan merah. Untuk mendapatkan informasi tersebut pengguna harus melakukan *scanning* suatu kode yang telah disediakan, setiap kali pengguna melakukan *scanning*, data pengguna seperti lokasi, nama dan kode identitas perangkat juga akan dikirim ke *server* untuk melakukan pelacakan pergerakan pengguna (Mehta, 2020).

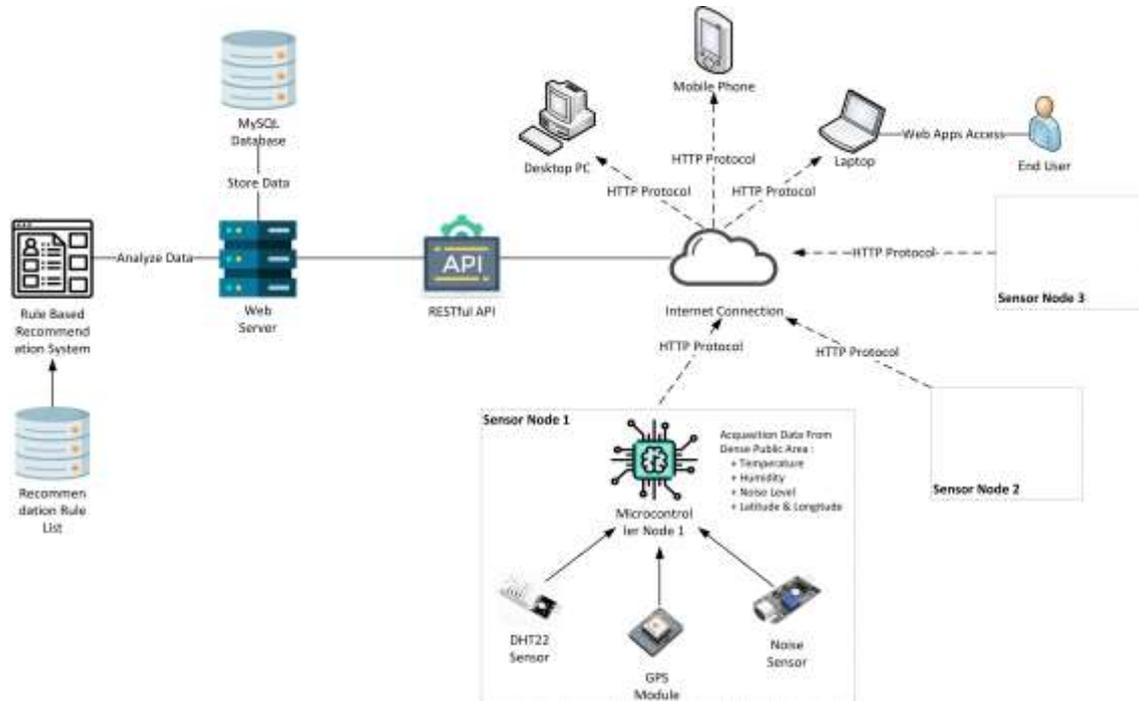
Aplikasi seperti *TraceTogether* dan *framework* Google/*Apple* dapat dimanfaatkan untuk penjelajahan riwayat kontak, akan tetapi tentunya akan menguras sumber daya baterai, karena aplikasi tersebut digunakan secara *realtime* untuk mengakses data riwayat penjelajahan kontak menggunakan koneksi *bluetooth*. Solusi lain yang ditawarkan adalah aplikasi *Alipay Sign a Color Code App* aplikasi ini telah memberikan solusi dari permasalahan diatas, akan tetapi pengguna harus merelakan data pribadinya diambil dan disimpan pada *server* setiap kali pengguna membutuhkan informasi tentang suatu tempat. Hal ini tentu berkaitan dengan masalah keamanan.

Dari beberapa aplikasi yang telah dijabarkan diatas, serta beberapa pendekatan solusi dari permasalahan yang telah dibahas, pada penelitian dibuat sebuah rancangan sistem monitoring pintar berbasis IoT yang berguna sebagai upaya mitigasi penularan COVID-19 di area padat publik. Sistem yang dirancang akan memonitor lingkungan sekitar dengan memanfaatkan beberapa sensor yang terpasang untuk membaca dan merekam data secara *realtime* seperti suhu, kelembaban, tingkat kebisingan dan data lokasi. Selain itu sistem yang dirancang juga menganalisis data suhu, kelembaban dan tingkat kebisingan sebagai parameter *input* untuk memberikan rekomendasi kepada pengguna.

## **METODE PENELITIAN**

Sistem monitoring pintar yang dirancang terdiri dari beberapa unit sistem yang saling terintegrasi. Adapun unit-unit tersebut adalah unit *sensor node*, *web server*, *web service* (RESTful API), *database service* (MySQL database), sistem rekomendasi pengguna (*user recommendation system*) dan aplikasi *client* berbasis *web* yang dapat

diakses di perangkat laptop, komputer *desktop* maupun perangkat *mobile* sebagai antarmuka sistem kepada pengguna akhir (*end user*). Adapun arsitektur sistem monitoring pintar yang dirancang dapat dilihat pada Gambar 1 berikut:



Gambar 1. Arsitektur Sistem Monitoring Pintar

Pada gambar diatas menunjukkan arsitektur sistem monitoring pintar yang terdiri dari beberapa unit sistem yang saling terintegrasi, unit *sensor node* berguna sebagai modul yang merekam dan memantau data lingkungan seperti data kelembaban, suhu, kebisingan dan lokasi. Data lingkungan yang diperoleh akan dikirim ke *web server* melalui *web service api* (RESTful API) yang dibuat. Selanjutnya data ini juga akan disimpan ke *database* MySQL untuk memastikan ketersediaan data secara *realtime* bagi pengguna akhir. Data suhu, kelembaban dan tingkat kebisingan akan dianalisis oleh sistem sebagai tolak ukur untuk memberikan rekomendasi kepada pengguna akhir.

Sistem rekomendasi bertugas untuk menganalisis data hasil pemantauan yang dikirim oleh *sensor node*. Data hasil pemantauan diproses oleh sistem rekomendasi untuk menemukan aturan yang cocok pada daftar aturan rekomendasi yang dibuat berdasarkan tingkat peringatan yang diperoleh. Acuan pembuatan aturan rekomendasi mengacu pada penelitian yang dilakukan oleh (Chan et al., 2011), Auler et all (Auler, Cássaro, da Silva, & Pires, 2020) dan Wu et all (Wu et al., 2020) yang melakukan analisis pengaruh *temperature* dan *humidity* terhadap resiko penyebaran COVID-19.

Sedangkan di sisi pengguna akhir terdapat aplikasi berbasis *web* sebagai antarmuka pengguna untuk mengakses sistem *monitoring*.

Selanjutnya hasil rancangan sistem monitoring pintar yang dibangun akan di uji untuk mendapatkan tingkat peforma sistem yang dibangun. Pengujian dilakukan terhadap unit sistem yang telah terintegrasi yaitu pengujian pengiriman paket data dan pengujian sampel acak dari aturan rekomendasi. Pengujian pengiriman paket data dilakukan dengan menghitung paket data yang hilang dari sensor node ke web server, persamaan yang digunakan adalah:

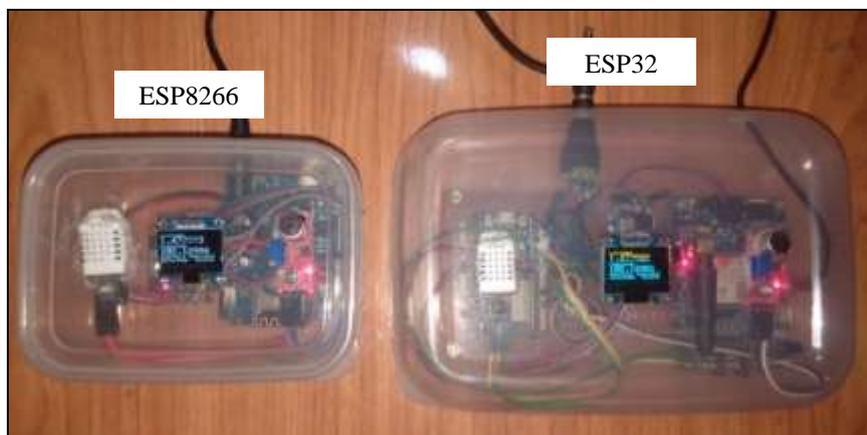
$$\text{Paket yang hilang} = \frac{\text{Jumlah paket yang dikirim} - \text{Jumlah paket yang diterima}}{\text{Jumlah paket yang diterima}} \times 100\% \dots\dots (1)$$

Sedangkan pengujian sampel acak aturan rekomendasi dilakukan dengan menghitung akurasi pengujian sampel acak melalui persamaan berikut:

$$\text{Akurasi pengujian sampel acak} = \frac{\text{Hasil uji yang benar} - \text{Hasil uji yang salah}}{\text{Total kasus uji}} \times 100 \% \dots\dots (2)$$

## HASIL DAN PEMBAHASAN

Berikut merupakan hasil *prototype* unit *sensor node* yang telah dikembangkan dapat dilihat pada Gambar 2.



Gambar 2. Hasil Prototype Sensor Node

Adapun *prototype sensor node* yang dikembangkan berjumlah 2 unit *sensor node*, masing-masing menggunakan modul WiFi ESP8266 dan modul WiFi ESP32 sebagai *controller*. Setelah mengembangkan perangkat *sensor node* tahap berikutnya ialah menguji pengiriman paket data antar *sensor node* ke web *server* menggunakan Arduino serial monitor dan riwayat data sensor. Berikut adalah hasil uji coba pengiriman data *sensor node* ESP8266 pada Arduino serial monitor:

```

LAT : 1.477631 LONG : 102.129723 TEMP : 32.4°C HUMI : 74.6% NOISE_LVL : 49db
HTTP Response code: 200, Length: 124, Counter: 2

{"error": 0,
 "message": "Data successfully added!"
}

LAT : 1.477628 LONG : 102.129768 TEMP : 32.4°C HUMI : 74.6% NOISE_LVL : 49db
HTTP Response code: 200, Length: 124, Counter: 3

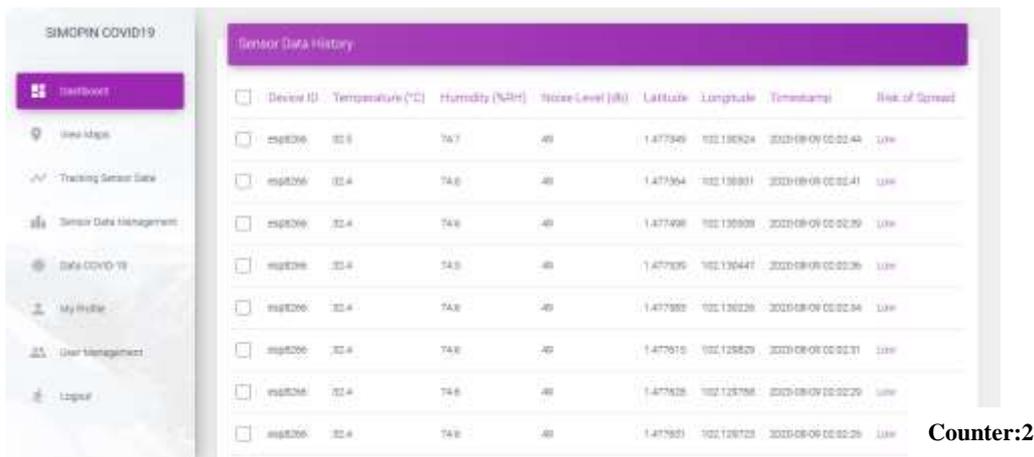
{"error": 0,
 "message": "Data successfully added!"
}

LAT : 1.477615 LONG : 102.129829 TEMP : 32.4°C HUMI : 74.6% NOISE_LVL : 49db
HTTP Response code: 200, Length: 124, Counter: 4

{"error": 0,
 "message": "Data successfully added!"
}
    
```

Gambar 3. Respon POST data ke *server* melalui RESTful API

Data sensor yang dikirim melalui metode post ke RESTful API dalam format JSON dan mengembalikan respon status pengiriman data juga dalam bentuk JSON dengan objek *error* dan *message*. Error bernilai 0 jika data berhasil diproses dan disimpan ke *database server*, bernilai 1 jika data tidak berhasil diproses dan disimpan kedalam *database server*. Selanjutnya data ditampilkan pada aplikasi web pengguna kedalam bentuk tabel seperti pada Gambar 4 berikut.



Gambar 4. Antarmuka Riwayat Sensor Data

Data sensor yang ditampilkan berurutan secara *descending* (waktu dari terbesar ke kecil) berdasarkan kolom data tanggal atau timestamp, sehingga data yang ditampilkan pada tabel selalu yang terkini. Hasil uji coba terhadap 30 kali pengiriman data dari *sensor node* ESP8266 dapat dilihat pada tabel berikut:

Tabel 1  
 Hasil Uji Coba Pengiriman Data Pada Perangkat ESP8266

Counter	Waktu Pengiriman	Latitude dan Longitude	Sukses Terkirim
0	2020-08-09 02:02:21	1.477673, 102.129646	Ya
1	2020-08-09 02:02:23	1.477643, 102.129631	Ya
2	2020-08-09 02:02:26	1.477631, 102.129723	Ya
3	2020-08-09 02:02:29	1.477628, 102.129768	Ya
4	2020-08-09 02:02:31	1.477615, 102.129829	Ya
...	...	...	Ya
29	2020-08-09 02:03:32	1.477291, 102.130081	Ya

Counter	Waktu Pengiriman	Latitude dan Longitude	Sukses Terkirim
30	2020-08-09 02:03:34	1.477291, 102.130081	Ya

Pengujian juga dilakukan untuk perangkat *sensor node* ESP32 dengan skenario yang sama yaitu 30 kali pengiriman data dari *sensor node* ke *web service api*, hasilnya dapat dilihat pada tabel berikut:

Tabel 2  
 Hasil Uji Coba Pengiriman Data Pada Perangkat ESP32

Counter	Waktu Pengiriman	Latitude dan Longitude	Sukses Terkirim
0	2020-08-09 03:35:39	1.477000, 102.129227	Ya
1	2020-08-09 03:35:43	1.477155, 102.129486	Ya
2	2020-08-09 03:35:45	1.477160, 102.129539	Ya
3	2020-08-09 03:35:48	1.477163, 102.129555	Ya
4	2020-08-09 03:35:51	1.477563, 102.129875	Ya
...	...	...	Ya
29	2020-08-09 03:36:54	1.476950, 102.129204	Ya
30	2020-08-09 03:36:56	1.476963, 102.129219	Ya

Berdasarkan pengujian pengiriman 30 paket data sensor yang telah dilakukan pada perangkat ESP8266 dan ESP32, terdapat 30 data yang diterima dan disimpan ke *database server* pada masing-masing skenario, sehingga persentase paket yang hilang dapat dihitung melalui persamaan berikut :

$$\begin{aligned}
 \text{Paket yang hilang} &= \frac{\text{Jumlah paket yang dikirim} - \text{Jumlah paket yang diterima}}{\text{Jumlah paket yang diterima}} \times 100\% \\
 &= \frac{(30-30)}{30} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Pengujian sampel acak dilakukan dengan memilih sampel data sensor (*temperature, humidity dan noise level*) secara acak kemudian melakukan *request POST* data ke *server* melalui aplikasi Postman untuk melihat apakah aturan rekomendasi bekerja sesuai dengan yang diharapkan. Hasil analisis terhadap sepuluh data uji yang dipilih secara acak beserta hasil aktual dan hasil yang diharapkan dapat dilihat pada tabel berikut.

Tabel 3  
 Kasus Uji Data Acak dan Hasil Analisis

Case id	Temperature (°C)	Humidity (%RH)	Noise Level (db)	Hasil Prediksi	Hasil yang diharapkan
1	27.5	49.7	56	High	High
2	26.3	67.1	59	High	High
3	29.2	78.2	58	Moderate	Moderate
4	32.6	79.6	43	Low	Low
5	28.8	81.4	57	Moderate	Moderate
6	34.7	77.8	45	Low	Low
7	25.9	60.3	51	Moderate	Moderate
8	35.6	83.7	54	Low	Low
9	27.8	78.9	67	High	High
10	33.4	90.5	42	Low	Low

Berdasarkan hasil pengujian sampel acak pada Tabel 3, maka diperoleh nilai akurasi pengujian acak yang dapat dihitung dengan persamaan berikut:

$$\begin{aligned} \text{Akurasi pengujian sampel acak} &= \frac{\text{Hasil uji yang benar} - \text{Hasil uji yang salah}}{\text{Total kasus uji}} \times 100 \% \\ &= \frac{10-0}{10} \times 100 \% \\ &= 100\% \end{aligned}$$

## SIMPULAN

Dari hasil pengujian yang telah dilakukan pada dua perangkat *sensor node* menunjukkan kinerja pengiriman data yang sangat baik dengan persentase paket yang hilang sebesar 0%. Selain itu pengujian data juga dilakukan dengan mengukur akurasi dari aturan rekomendasi yang telah dibuat dengan data dipilih secara acak sebanyak 10 data, dimana hasil akurasi yang diperoleh adalah sebesar 100%. Maka dapat disimpulkan bahwa rancangan yang telah dibuat dapat bekerja dengan baik dan dapat diimplementasikan. Saran pengembangan dari penelitian ini adalah perlunya kajian penerapan protokol lain seperti misalnya protokol MQTT dan *Socket* dalam melakukan pengiriman data ke *web server* sehingga kinerja layanan jaringan dapat dimaksimalkan.

## DAFTAR PUSTAKA

- Auler, A. C., Cássaro, F. A. M., da Silva, V. O., & Pires, L. F. (2020). Evidence that high temperatures and intermediate relative humidity might favor the spread of COVID-19 in tropical climate: A case study for the most affected Brazilian cities. *Science of the Total Environment*, 729. Retrieved from <https://doi.org/10.1016/j.scitotenv.2020.139090>
- Bay, J., Kek, J., Tan, A., Hau, C. S., Yongquan, L., Tan, J., & Quy, T. A. (2020). *BlueTrace : A privacy-preserving protocol for community-driven contact tracing across borders*. Singapore.
- Chan, K. H., Peiris, J. S. M., Lam, S. Y., Poon, L. L. M., Yuen, K. Y., & Seto, W. H. (2011). The effects of temperature and relative humidity on the viability of the SARS coronavirus. *Advances in Virology*, 2011. Retrieved from <https://doi.org/10.1155/2011/734690>
- Google, & Apple. (2020). Contact tracing. Diambil 22 April 2020, dari Retrieved from <https://www.apple.com/covid19/contacttracing/>
- Government Technology Agency Singapore. (2020). TraceTogether. Diambil 22 April 2020, Retrieved from <https://www.tech.gov.sg/media/technews/tracetogether-behind-the-scenes-look-at-its-development-process>
- Mehta, I. (2020). China's Coronavirus detection app. Diambil 22 April 2020, Retrieved from <https://thenextweb.com/china/2020/03/03/chinas-covid-19-app-reportedly-color-codes-people-and-shares-data-with-cops/>
- Wu, Y., Jing, W., Liu, J., Ma, Q., Yuan, J., Wang, Y., ... Liu, M. (2020). Effects of temperature and humidity on the daily new cases and new deaths of COVID-19 in 166 countries. *Science of the Total Environment*, 729, 1–7. Retrieved from <https://doi.org/10.1016/j.scitotenv.2020.139051>