



## Chatbot LINE untuk monitor perkembangan kasus COVID-19 di Indonesia menggunakan PHP dan MySQL

Rizky Parluka<sup>a,1</sup>, Sunu Ilham Pradika<sup>a</sup>, Amir Muhammad Hakim<sup>a</sup>, Kholilul Rachman Nur Manab<sup>a</sup>

<sup>a</sup>Jurusan Informatika Universitas Pembangunan Nasional “Veteran” Jawa Timur, Jl. Rungkut Madya No.1, Surabaya, Jawa Timur 60294, Indonesia

<sup>1</sup>E-mail: [rizkyparlika.if@upnjatim.ac.id](mailto:rizkyparlika.if@upnjatim.ac.id)

### INFO ARTIKEL

#### Riwayat artikel:

Diajukan pada 31 Mei 2020

Direvisi pada 19 Juli 2020

Disetujui pada 10 Oktober 2020

Tersedia daring pada 15 November 2020

#### Kata kunci:

COVID-19, API, chatbot, LINE messaging API.

#### Keywords:

COVID-19, API, chatbot, LINE messaging API.

### ABSTRAK

Seiring dengan perkembangan teknologi yang telah banyak mempengaruhi bidang kehidupan, khususnya di bidang informasi. Banyak informasi yang bertebaran dengan sangat cepat contohnya seperti berita akhir-akhir ini yang mendunia yaitu *Coronavirus Disease 2019* (COVID-19). COVID-19 adalah penyakit yang disebabkan oleh virus Corona tipe baru. Saat ini berbagai media informasi sedang gencar-gencarnya menyebarkan informasi mengenai COVID-19, sehingga sebagai penelitian dalam menangani virus ini peneliti membangun aplikasi chatbot dengan memanfaatkan LINE Messaging API sebagai platform dan PHP language sebagai bahasa pemrograman backend, serta MySQL sebagai penyimpanan basis data. Tujuan yang diharapkan dalam pembuatan aplikasi ini untuk memberikan berita informasi penyebarluasan *Coronavirus Disease 2019* di Indonesia. Dengan melakukan implementasi dari sistem yang telah dirancang, maka berhasil dibuat sebuah sistem yang dapat melakukan pengambilan data dari API yang memberikan data kasus COVID-19 di Indonesia dan sebuah chatbot yang dapat berinteraksi dengan pengguna LINE.

### ABSTRACT

Along with the development of technology which has greatly influenced life, especially in the field of information, a lot of information spreads very quickly such as the recent news worldwide, namely *Coronavirus Disease 2019* (COVID-19). COVID-19 is a disease caused by a new type of Coronavirus. Currently various information media are incessantly spreading information about COVID-19. Dealing with this issue, researchers are building chatbot applications by utilizing the LINE messaging API as a platform and PHP language as a backend programming language, and MySQL as database storage. The expected goal in making this application is to provide news information on the distribution of *Coronavirus Disease 2019* in Indonesia. By implementing the system that has been designed, a successful system can be made to retrieve data from the API that provides COVID-19 case data in Indonesia and a chatbot that can interact with LINE users.

Tersedia pada: <http://dx.doi.org/10.36055/tjst.v16i2.8240>.

## 1. Pendahuluan

*Coronavirus Disease 2019* (Covid-19) adalah penyakit yang disebabkan oleh virus Corona tipe baru. Rata-rata penderita COVID-19 akan mengalami gangguan pernafasan tingkat sedang hingga akut sehingga penderita memerlukan penanganan khusus oleh tenaga kesehatan, yakni dengan perawatan intensif disertai isolasi. COVID-19 sangat mudah menular antar individu. Salah satu cara terbaik untuk mencegah penularannya adalah dengan menyebarkan informasi mengenai COVID-19 seperti bagaimana virus ini menular dan apa penyebabnya serta cara bagaimana melindungi diri sendiri agar tidak tertular [1].

Media berita mempunyai peran untuk menyebarkan informasi-informasi penting maupun populer [2]. Contohnya saat ini berbagai media yang sedang gencar-gencarnya menyebarkan informasi mengenai COVID-19 yang diantaranya meliputi berita mengenai COVID-19 dan tips pencegahan agar tidak tertular hingga statistik kasus penderita COVID-19. Jika diamati, media yang dapat digunakan untuk menyebarluaskan informasi mengenai COVID-19 tidak hanya menggunakan media berita melainkan juga dapat dilakukan dengan menggunakan media pengiriman pesan instan (*chatting*) yang disediakan oleh aplikasi pengiriman pesan instan. Salah satu contoh media pengiriman pesan instan tersebut adalah LINE [3-6].



LINE merupakan *platform* komunikasi yang menawarkan layanan *chatting*, *voice call* dan *video call* [7, 8]. LINE juga menyediakan fitur lain seperti penyediaan konten informasi, hiburan dan bisnis [9]. LINE menyediakan sebuah API yakni *LINE messaging API* yang dapat digunakan oleh pengembang untuk mengembangkan *chatbot* [3-6]. *Chatbot* adalah perangkat lunak yang dibangun untuk mensimulasikan percakapan dengan pengguna melalui perantara internet [3-5, 10-13]. *chatbot* dapat digunakan sebagai media berbagi informasi yang tentunya dibutuhkan oleh pengguna [14, 15]. Khususnya untuk berbagi informasi yang paling banyak dicari pada sekarang ini yakni informasi mengenai statistik kasus COVID-19.

Oleh karena itu, pada penelitian ini penulis mengembangkan sebuah *chatbot* yang dapat digunakan oleh pengguna melalui aplikasi *chatting* yang dimiliki oleh LINE untuk meminta informasi statistik kasus COVID-19 di Indonesia secara *realtime*. Sistem dibangun menggunakan bahasa pemrograman PHP dan basis data MySQL. Dengan menggunakan kombinasi keduanya dapat dibangun sebuah *bot* yang dapat berinteraksi dengan pengguna [12]. Pengguna dapat mengirimkan pesan berupa perintah kepada *chatbot*, lalu dengan segera *chatbot* akan memberikan informasi tersebut melalui *chat*.

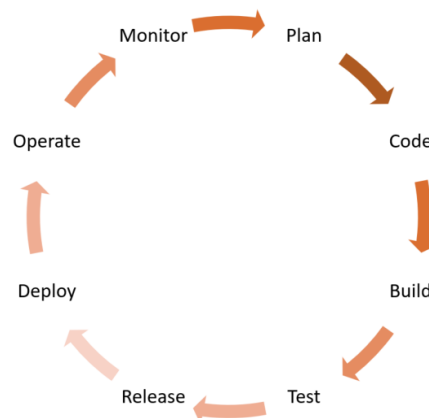
Tujuan dari adanya *chatbot* ini adalah memudahkan pengguna yang ingin mencari informasi mengenai statistik kasus COVID-19 di Indonesia yakni dengan mengirimkan pesan perintah kepada *chatbot* untuk meminta informasi tersebut. Adapun sumber data statistiknya didapatkan dari sumber yang kredibel dan selalu diperbaharui setiap harinya, yakni dari API yang disediakan oleh Kawal Corona. Kawal Corona merupakan sebuah situs *online* yang menyediakan data statistik kasus COVID-19 di Indonesia dan di seluruh dunia. Situs ini dikembangkan oleh seorang relawan dari komunitas Ethical Hacker Indonesia [16].

Beberapa penelitian yang terkait dengan *chatbot* LINE untuk memonitor penyebaran berita *hoax* pada *platform* LINE yakni penelitian yang dilakukan oleh Arita dan Finandhita (2019) yang mampu menerapkan *chatbot* LINE untuk memonitor penyebaran berita *hoax* dengan memprediksinya menggunakan metode *rule based* [9]. Pada tahun 2020 juga terdapat penelitian serupa yaitu penerapan *chatbot* LINE oleh Lusita *et al.* (2020), yaitu membuat aplikasi *bot* badan administrasi akademik dan kemahasiswaan (BAAK) yang digunakan untuk memudahkan informasi administrasi mahasiswa [7]. Disisi lain, pada penelitian yang dilakukan oleh Wirasandi *et al.* (2019) yang mampu menerapkan *chatbot* LINE untuk membuat aplikasi layanan pesan antar makanan di kampus dengan menggunakan metode *iterative incremental* [13].

## 2. Metodologi Penelitian

### 2.1. Software Development Life Cycle (SDLC)

*Software development life cycle* (SDLC) atau yang dalam Bahasa Indonesia siklus hidup pengembangan perangkat lunak yang diterapkan pada sistem *chatbot* ini adalah *continuous integration*, *continuous delivery* (CI/CD), dan *continuous development*, yakni sebuah siklus dimana perancangan, pembangunan, pengembangan, implementasi, dan evaluasi dilakukan secara terus menerus [17]. Gambar 1 berikut adalah ilustrasi dari SDLC CI/CD.

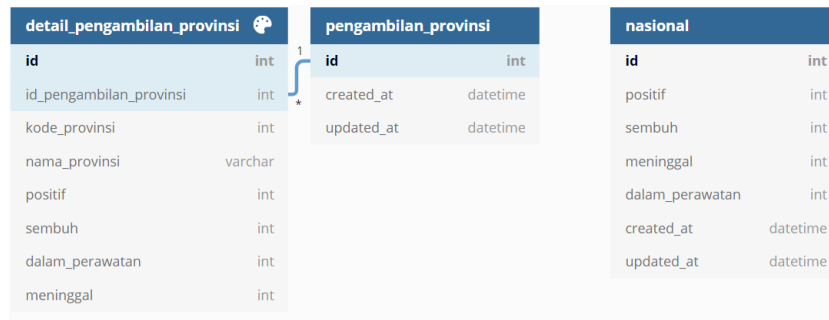


Gambar 1. SDLC CI/CD.

Siklus ini dimulai dari tahap perencanaan apa saja yang dibutuhkan untuk membangun *software*. Kemudian dilanjutkan membangun aplikasi dengan melakukan pengkodean. Setelah berhasil dibangun, maka langsung dilakukan pengujian fitur dan apabila memenuhi perencanaan awal maka akan dilakukan rilis ke publik dan segera dilakukan *deploy* agar segera dapat dinikmati oleh pengguna. Pengembang dalam hal ini melakukan pengoperasian dan melakukan pengawasan terhadap sistem yang telah dibuat.

### 2.2. Basis Data

Gambar 2 memuat desain basis data yang digunakan pada sistem server yang digunakan sebagai *webhooks* untuk *LINE messaging API*. Pada gambar tersebut terdapat 3 tabel yaitu nasional, pengambilan provinsi, dan detail pengambilan provinsi. Tabel nasional digunakan untuk menyimpan data kasus COVID-19 secara nasional di Indonesia. Lalu, pada pengambilan tabel provinsi perlu dilakukan pemecahan tabel menjadi dua yakni pengambilan provinsi dan detail pengambilan provinsi. Karena setiap sekali pengambilan data provinsi maka disitu terdapat banyak provinsi, sehingga perlu membuat sebuah tabel baru untuk menghindari relasi *many-to-many* pada basis data. Basis data yang digunakan pada penelitian ini adalah MySQL. Data yang didapatkan beberapa saling terhubung sehingga menggunakan *relational database management system* (RDBMS) merupakan pilihan yang tepat [18, 19].



Gambar 2. Desain basis data.

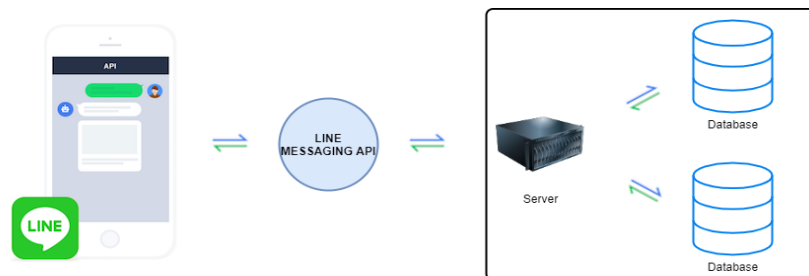
### 2.3. Pengambilan Data API



Gambar 3. Arsitektur sistem pengambilan data.

Data diambil dari situs [api.kawalcorona.com](http://api.kawalcorona.com) dimana pada situs tersebut terdapat data kasus COVID-19 baik dalam negeri dan luar negeri. Situs tersebut juga memuat data kasus COVID-19 pada setiap provinsi di Indonesia. Program yang telah dibuat akan diletakkan pada server kemudian dibuat sebuah *cron jobs* yaitu proses eksekusi *script* secara otomatis oleh server dengan siklus menit, jam, minggu, atau bulan [20]. Dengan memanfaatkan *cron jobs* maka data yang ada pada basis data dapat diperbarui secara berkala sehingga kasus COVID-19 yang tersedia pada *bot* adalah data terakhir yang didapatkan dari API. Pada server, dibuat sebuah *cron jobs* yang berjalan setiap 5 menit sekali. Jadi, data kasus COVID-19 diperbarui setiap 5 menit sekali dengan memanggil *script* yang ada.

### 2.4. Arsitektur Sistem



Gambar 4. Arsitektur sistem bot.

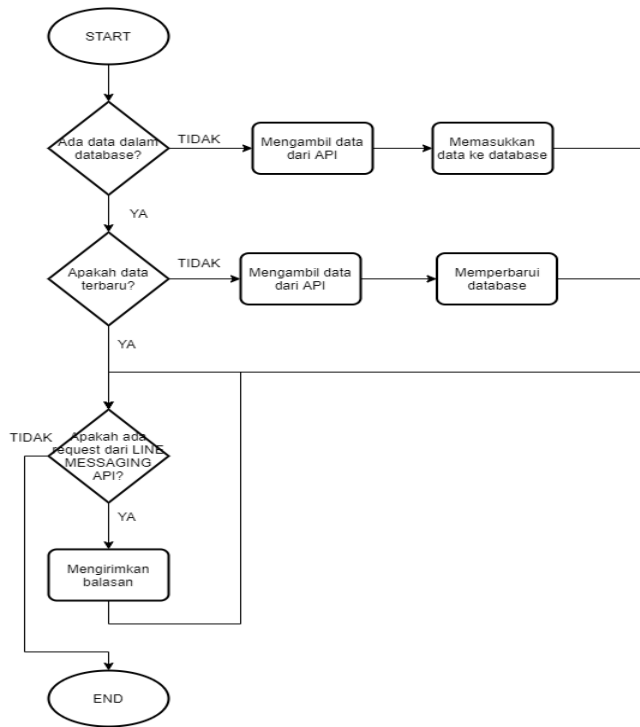
LINE *messaging API* memungkinkan data untuk dikirimkan antara *server bot* yaitu *hosting* dan *platform LINE*. Permintaan dikirim melalui *HTTPS request* dalam format *JSON* [21]. Cara kerjanya adalah sebagai berikut:

- Pengguna mengirim pesan ke official account LINE yang digunakan sebagai *bot*.
- Platform LINE mengirimkan *event webhook* ke *webhook URL* yang diisikan pada pengaturan LINE *messaging API* ke *server bot*.
- Berdasarkan *event webhook* yang diterima, *server bot* merespons pengguna melalui *platform LINE*.

### 2.5. Alur Sistem

Gambar 5 memuat alur sistem yang dijalankan pada *server*. Sistem berperan dalam melakukan pengambilan data dari API dan menentukan apa balasan yang harus diberikan kepada pengguna melalui LINE *messaging API*. Pada peran untuk pengambilan data dari API dilakukan dengan cara melakukan pengecekan. Apabila tidak terdapat data dalam basis data maka masukkan data baru. Namun, apabila terdapat data dalam basis data, maka lakukan pengecekan dengan membandingkan data dalam basis data dengan data yang diambil dari API. Jika terdapat pembaruan data dari API maka perbarui data.

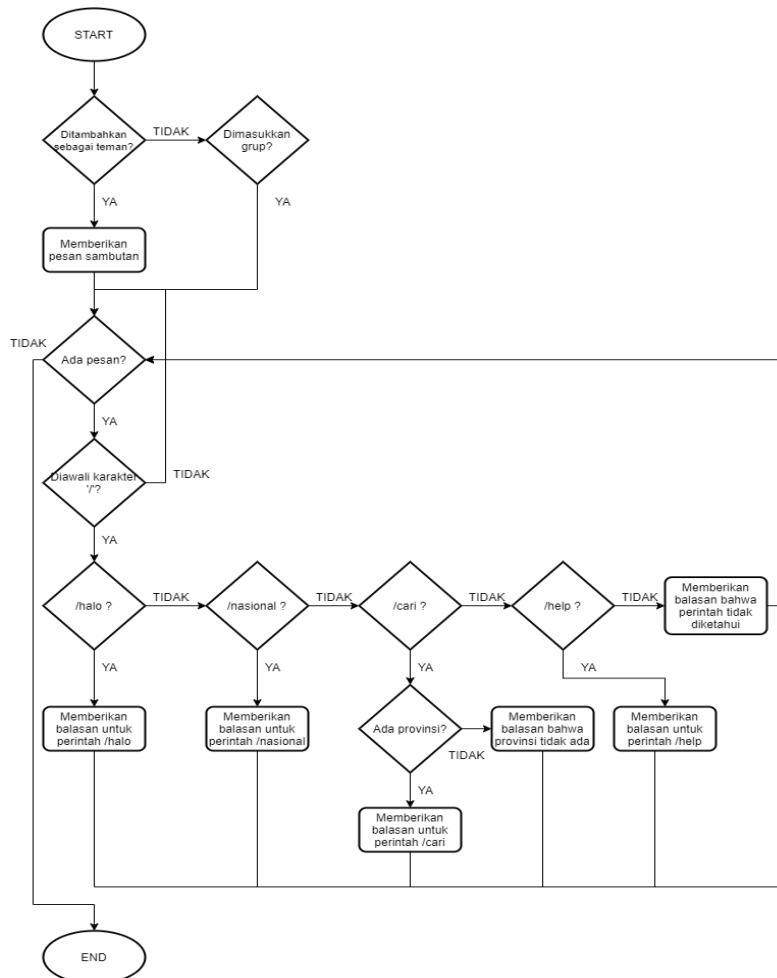
Peran selanjutnya yaitu menentukan balasan apa yang harus diberikan kepada pengguna. Langkah pertama yaitu melakukan pengecekan apakah ada *HTTPS request* yang diterima dari LINE *messaging API*. Selanjutnya, jika terdapat *request* maka akan dilakukan pengecekan apakah itu termasuk dalam *event* pesan. Terakhir, jika itu adalah *event* pesan maka akan ditentukan balasan yang sesuai yang akan dijelaskan pada alur *bot*.



Gambar 5. Alur sistem pada server.

2.6. Alur Bot

Alur dari bot yang dibuat termuat pada Gambar 6. Sebelum dapat berinteraksi dengan bot, diharuskan menambahkan bot yang merupakan sebuah official account pada LINE sebagai teman. Kemudian setelah menambahkan teman, bot baru bisa digunakan. Setelah ditambahkan sebagai teman, maka bot akan memberikan sambutan awal atau greeting message. Setelah itu, maka bot akan menunggu apakah terdapat pesan yang dikirimkan kepadanya.

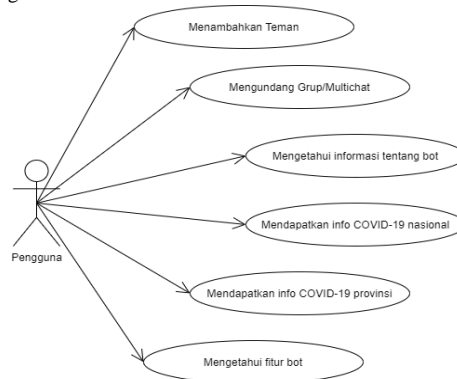


Gambar 6. Alur bot saat berinteraksi dengan pengguna.

Selanjutnya, apabila terdapat pesan maka akan dilakukan pengecekan untuk mengetahui karakter pertama dari sebuah pesan tersebut apakah “/”, karena untuk dapat menggunakan fitur yang ada pada *bot* setiap pesan yang dikirimkan harus diawali “/”. Apabila kondisi bernilai benar, maka akan melakukan pengecekan kembali berupa kata selanjutnya apakah termasuk “/halo”, “/nasional”, “/cari”, dan “/help”. Perintah “/halo” digunakan untuk mendapatkan perkenalan dari *bot* mengenai fungsinya. Kata “/nasional” digunakan untuk mendapatkan info kasus COVID-19 yang ada di Indonesia (skala nasional). Kata “/cari” sendiri harus mendapatkan tambahan berupa nama provinsi, misalkan papua, maka akan menjadi “/cari Papua”. Perintah “/cari” digunakan untuk mendapatkan info kasus COVID-19 yang ada di tiap provinsi di Indonesia. Terakhir, “/help” adalah perintah untuk mengetahui perintah/fitur apa saja yang tersedia pada *bot*.

### 3. Hasil dan Pembahasan

Berikut adalah *use case* dari sistem yang dibangun.



Gambar 7. Use case diagram.

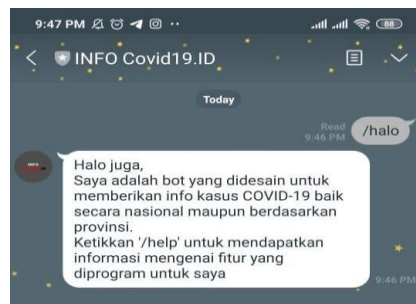
Didapatkan sebuah *use case* dimana pengguna dapat melakukan beberapa aksi yang dilakukan untuk berinteraksi dengan *bot*. Diantaranya adalah menambahkan teman karena ada fitur dari LINE untuk pengaturan tentang penambahan teman sehingga dapat dibatasi atau tidak bisa ditambahkan, mengundang *bot* ke dalam grup/*multichat*. Kemudian beberapa interaksi yang terkait dengan fitur utama *bot* yakni:

1. Mengetahui informasi tentang *bot*.
2. Mendapatkan informasi kasus COVID-19 skala nasional.
3. Mendapatkan informasi kasus COVID-19 skala provinsi.
4. Mengetahui fitur *bot*.



Gambar 8. Greeting message dari bot.

Pada Gambar 8 adalah contoh tampilan fitur bawaan dari LINE dimana setiap ada pengguna baru yang menambahkan teman sebuah *official account* maka akan diberikan sebuah *greeting message*.



Pada Gambar 9 memuat sebuah perintah untuk mengetahui informasi tentang *bot*.



**Gambar 10. Informasi kasus COVID-19 nasional.**

Pada Gambar 10 adalah contoh tampilan sebuah perintah untuk mengetahui bagaimana perkembangan kasus COVID-19 pada skala nasional.



**Gambar 11. Informasi kasus COVID-19 provinsi.**

Pada Gambar 11 adalah contoh tampilan sebuah perintah untuk mengetahui bagaimana perkembangan kasus COVID-19 pada skala provinsi. Seluruh 34 provinsi dapat diakses melalui *bot* ini.



**Gambar 12. Informasi mengenai fitur *bot*.**

Pada Gambar 12 adalah contoh tampilan sebuah perintah untuk mengetahui fitur yang ada pada *bot*. Di sana juga terdapat bagaimana perkembangan versi dari *bot*.

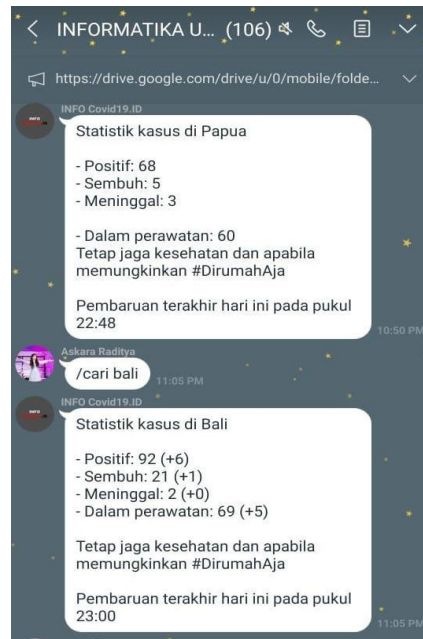


**Gambar 13. Penanganan *error* berupa perintah yang tidak dikenal.**



**Gambar 14. Penanganan *error* apabila provinsi tidak ada dalam basis data.**

Pada Gambar 13 dan Gambar 14 adalah contoh tampilan sebuah salah satu fitur *bot* yakni penanganan terhadap *error* baik perintah yang tidak dikenal maupun tidak ditemukannya provinsi.



**Gambar 15. Mengundang bot ke dalam grup.**

Pada Gambar 15 adalah contoh tampilan ketika *bot* dimasukkan ke dalam grup kemudian semua orang dalam grup menggunakan semua fitur yang tersedia pada *bot*. Berikut pada Tabel 1 adalah hasil pengujian *black box* yang telah dilakukan.

**Tabel 1. Pengujian dengan metode *black box*.**

No	Pengujian	Keterangan
1	Memberikan greeting message	Berhasil
2	Memberikan error handling atau penanganan error	Berhasil
3	Memberikan informasi mengenai kasus COVID-19 tingkat nasional.	Berhasil
4	Memberikan informasi mengenai kasus COVID-19 tingkat provinsi.	Berhasil
5	Dapat dimasukkan ke dalam grup	Berhasil
6	Dapat ditambahkan sebagai teman	Berhasil
7	Memberikan informasi mengenai bot	Berhasil
8	Memberikan informasi mengenai fitur dan versi	Berhasil

#### 4. Kesimpulan dan Saran

*Chatbot* ini dapat digunakan oleh pengguna untuk mencari informasi statistik kasus COVID-19 di Indonesia. Pengguna dapat menggunakan perintah-perintah yang telah disediakan untuk melakukan pencarian informasi, baik pada lingkup nasional maupun provinsi. Fitur yang dimiliki oleh *chatbot* ini untuk mencari informasi antara lain sebagai berikut:

1. Informasi statistik nasional, yakni total keseluruhan kasus COVID-19 di Indonesia.
2. Informasi statistik kasus COVID-19 di provinsi tertentu, dilakukan dengan cara melakukan pencarian berdasarkan nama provinsi.

Pada bagian hasil dan pembahasan, tampak pengujian telah dilakukan oleh penulis untuk menjalankan seluruh fitur dari *chatbot*, yakni dengan mengirimkan berbagai pesan perintah yang tersedia kepada *chatbot*. Setelah itu penulis menerima serta mengamati respon balik dari *chatbot*. Kemudian hasilnya *chatbot* ini telah mampu memberikan respon yang sesuai dengan pesan perintah yang dikirimkan.

*Chatbot* ini masih dalam tahap penelitian, akan tetapi berdasarkan hasil program serta pengujian yang telah diperlihatkan sebelumnya menunjukkan bahwa *chatbot* ini telah layak dirilis serta digunakan oleh masyarakat umum untuk mendapatkan informasi statistik kasus COVID-19 di Indonesia dengan mudah dan cepat. Dikarenakan *chatbot* ini menyajikan informasi statistik dalam bentuk teks, maka untuk pengembangan selanjutnya penulis menyarankan agar ditambahkan fitur baru, yakni fitur untuk menghitung persentase pertumbuhan eksponensial harian serta menyajikan persentase tersebut ke pengguna. Penyajian persentase ini berguna untuk memudahkan pengguna melihat laju pertumbuhan dari hari ke hari.

## Ucapan Terima Kasih

Penulis berterima kasih kepada seluruh pihak yang telah memberikan saran, masukan, dan dukungan baik berupa fisik maupun non fisik sehingga penelitian ini bisa dilakukan hingga selesai, diantaranya adalah Mas Teguh Aprianto selaku pengembang layanan API Kawal Corona untuk objek penelitian. Tidak lupa penulis berterima kasih kepada para *reviewer* yang memberikan masukan yang tidak ternilai harganya untuk makalah ini sehingga menjadi baik.

## DAFTAR PUSTAKA

- [1] WHO. Coronavirus. (2020). Retrieved from <https://www.who.int/health-topics/coronavirus>.
- [2] Parluka, R., & Pratama, A. (2019). Aplikasi penampil berita *realtime* berbasis *bot* Telegram menggunakan API WEB (APBR versi 1.0). *Prosiding Seminar Nasional SANTIKA Ke-1 2019*, pp. 17-20.
- [3] Chandra, Y. I., & Kosdiana. (2019). Rancang bangun aplikasi *chat bot line* menggunakan pendekatan *agile process* dengan model *extreme programming* berbasis web. *Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK)*, vol. 3, no. 1, pp. 149-160.
- [4] Susilayasa, I. M. A., & Atitama, I. G. O. G. (2017). Implementasi *line chat bot* rekomendasi wisata menggunakan *platform as a service*. *Prosiding Seminar Nasional Teknologi Informasi & Aplikasinya (SNATIA)*, 29-33.
- [5] Jati, D. P., & Maarif, M. R. (2018). The development of chatbot application on line messaging platform for customer service in Jogja sewa kamera. *Compiler*, vol. 7, no. 2, pp. 91-98.
- [6] Prayoga, A., & Nurjayadi. (2018). *Bot* permainan tebak gambar pengenalan ada istiadat riau pada aplikasi LINE dengan metode *webhooks*. *Jurnal Teknologi Terpadu (JTT)*, vol. 6, no. 2, pp. 161-167.
- [7] Lusita, M. D., Humianingsih, H., & Rihyanti, E. (2020). Aplikasi *bot* akademik BAAK STMIK Jakarta STI&K *platform* Line messenger menggunakan *go languages*. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 3, no. 1, pp. 1-8.
- [8] LINE Corporation. (2019). *Line Official Account Operational Guide*. Retrieved from <https://www.linebiz.com/system/files/id/LINE>.
- [9] Arita, A., & Finandhita, A. (2019). Pembangunan *chatbot* untuk menampilkan berita *hoax* pada *platform* Line menggunakan metode *rule based* [skripsi]. E-Library Unikom (Universitas Komputer Indonesia).
- [10] Ananda, D. R., Imamah, F., Andre, S. Y. M., & Dores, D. (2018). Aplikasi *chatbot* (*milki bot*) yang terintegrasi dengan web CMS untuk *customer service* pada UKM MINSU. *Jurnal Cendikia*, vol. 16, no. 1, pp. 100-106.
- [11] Wijaya, M., Junaedy & Arfandy, H. (2019). Perancangan *chatbot* untuk informasi penerimaan mahasiswa baru pada STMIK Kharisma Makassar. *Kharisma Tech*, vol. 14, no. 1, pp. 14-23.
- [12] Parluka, R., & Pratama, A. (2019). Aplikasi mesin penjawab pesan berbasis *bot* Telegram, PHP, dan MySQL. *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, vol. 14, no. 3, pp. 1-9.
- [13] Wirasandi, A., Gumilang, S. F. S., & Hasibuan, M. A. (2019). Integrasi LINE *bot* layanan pesan antar makanan “di kampus” menggunakan LINE *front-end framework* (LIFF) dengan metode *iterative incremental*. *E-Proceeding of Engineering*, vol. 6, no. 2, pp. 8019-8026.
- [14] Parluka, R., Hutama, S. H., Gallanta, R. A., Nabilah, Q. J., & Nusari, A. (2019). Studi komparatif implementasi *push message* pada media sosial secara gratis. *E-Narodroid*, vol. 5, no. 2, pp. 44-55.
- [15] Parluka, R., Miftakhoneki, S., Fernanda, R. A., Ramadhan, F. D., & Andry S, F. (2020). Program menghitung rumus bangun datar dan bangun ruang menggunakan fasilitas *bot* Telegram. *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, vol. 15, no. 1, pp. 50-58.
- [16] Kawal Corona. Coronavirus Global & Indonesia Live Data. (2020). Retrieved from <https://kawalcorona.com/>
- [17] Azeri, I. (2017). What Is CI/CD?. Retrieved from <https://dzone.com/articles/what-is-cicd>.
- [18] Parluka, R., Prasetya, L. M. I., Putra, H. R., Satria, V. H., & Pralas, F. H. (2019). Membangun *login* terenkripsi menggunakan *bot* Telegram dan *database* MySQL. *Prosiding Seminar Nasional SANTIKA Ke-1 2019*, pp. 196-202.
- [19] Warman, I., & Ramdaniyasyah, R. (2018). Analisis perbandingan kinerja *query database management system* (DBMS) antara Mysql 5.7.16 Dan Mariadb 10.1. *Jurnal Teknoif*, vol. 6, no. 1, pp. 32-41.
- [20] Development & Security. Kenali Apa itu Cron Jobs?. (2016). Retrieved from <https://idcloudhost.com/kenali-apa-itu-cron-jobs/>.
- [21] LINE Corporation. (2020). Messaging API overview. Retrieved from <https://developers.line.biz/en/docs/messaging-api/overview/>.