TRIAC
Jurnal Teknik Elektro dan Komputer
Terakreditasi sinta 4
ISSN 2615-5788 (Online)
ISSN 2615-7764 (Print)
Publisher: Teknik Elektro Universitas Trunojoyo Madura

USER

Username
Password
Remember me
Login

GUIDELINES (PDF, WORD)

Article Template

STATEMENT OF ORIGINALITY

Statement Of Originality

TOOLS

MENDELEY

PLAGIARISM CHECKER

turnitin

ASSOCIATION

ISSN

Print
2615-7764
Online
2615-5788

STATISTICS

Visitors
ID 40,879 JP 53

Vol 9, No 3 (2022)

November 2022

DOI: https://doi.org/10.21107/triac.v9i3

Table of Contents

Engineering

IDENTIFIKASI PENYEBAB CACAT PADA HASIL PENGELASAN DENGAN IMAGE PROCESSING MENGGUNAKAN METODE YOLO — PDF (BAHASA INDONESIA) 107-112
Agus Khumaidi, Rizal Lucky Pradana

Aplikasi Bot Telegram Pada Sistem Presensi dan Pengukuran Suhu Tubuh Berbasis IoT — UJS_1_22 PDF 113-121
Rachmat Trishardian, Ari Fadli, Muhammad Syaiful Aliim, Retno Supriyanti, Yogi Ramadhani

Analisis Penggunaan Teknik Load Balancing untuk Optimalisasi Jaringan — PDF (BAHASA INDONESIA) 122-124
Ahmad Irsyad Amru Rosyidi, Rima Tri Wahyuningrum

Simulasi Kendali Sistem Suspensi Aktif Kendaraan Roda Empat Menggunakan Metode Full State Feedback dan PID — PDF 125-129
Monika Faswia Fahmi, Deni Tri Laksono

Pengimplementasian Metode Fuzzy Logic pada Kontrol Rumah Jamur Otomatis Berbasis Node-RED — PDF (BAHASA INDONESIA) 130-134
Fransiskus Belicoy Awi, Abd Rabi, Wahyu Dirgantara

Kontrol Prototipe Ruang Monitoring Kesehatan Berbasis Node-Red — PDF (BAHASA INDONESIA) 135-140
Millenia Zwi Sabatina Sirait, Elta Sonalitha, Wahyu Dirgantara

Jurnal Teknik Elektro dan Komputer TRIAC by Universitas Trunojoyo Madura is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

ISSN 2615-5788 (Online) and ISSN 2615-7764 (Print)

Indexed by:

Article Submission
Focus and Scope
CONTACT US

JOURNAL CONTENT

Search
Search Scope
All
Search

Browse
- By Issue
- By Author
- By Title
- Other Journals

INFORMATION
- For Readers
- For Authors
- For Librarians

EBOOK Free Downloads
e-journals

00110238
View My Statistics
statcounter
Google Analytics

TUTORIAL

TUTORIAL MENDELEY

NOTIFICATIONS

- View
- Subscribe

LANGUAGE

Select Language

English

Submit

KEYWORDS

Arduino Automation Academy, video, cameraman, auto-tracking. Bahan bakar Deque IMU Mapping Maximum Likelihood Detection Minimum Mean Square Error Mobile Robot Panel Surya, Heatsink, Reflektor Path Planning Physical Distancing, Sensor GY-906, Sensor IR Obstacle, Telegram Sensor Ultrasonik Telur asin, asam cuka, suhu, Proportional Integral Derivative, Zeigler-Nichols V-BLAST Webots YOLO CNN, Causes of Welding Defects, Ampere Zero Forcing bio-etanol listrik sensor laser, lebar rel, rel kereta api, sensor jarak, kalman filter.

APLIKASI *BOT* TELEGRAM PADA SISTEM PRESENSI DAN PENGUKURAN SUHU TUBUH BERBASIS IOT

Rachmat Trishardian, Ari Fadli, M Syaiful Aliim, Retno Supriyanti, Yogi Ramadhani
Teknik Elektro, Universitas Jenderal Soedirman, Jl. Raya Mayjen Sungkono No.KM 5, Purbalingga, 53371, Indonesia
E-mail: arifadli@unsoed.ac.id, muhammad.syaiful.aliim@unsoed.ac.id,
retno_supriyanti@unsoed.ac.id, yogi.ramadhani@unsoed.ac.id

Abstrak— Virus SARS-CoV-2 merupakan yang menyebabkan penyakit menular COVID-19. Berkaca pada data yang tersedia pada sumber terbuka worldometers.info pada 30 Januari 2022 ditemukan 375.001.247 kasus virus Covid-19 di seluruh dunia. Sehingga hal tersebut membuktikan bahwa tingginya kasus virus Covid-19 ini diperlukan upaya untuk mencegah penyebarannya. Tidak hanya dengan menerapkan perilaku hidup sehat, upaya pencegahan penyebaran virus tersebut dilakukan melalui hadirnya inovasi teknologi. Diantara Inovasi teknologi tersebut hadir dalam bentuk penggunaan bot Telegram pada sistem presensi dan pengukuran suhu tubuh berbasis internet of things (IoT) yang berfungsi sebagai media pemantauan data, pelaporan, dan notifikasi. Kehadiran Bot Telegram yang difungsikan untuk menjalankan perintah dan memberikan respon sesuai dengan permintaan. Dalam penelitian ini, *bot* Telegram dibuat dengan menggunakan bahasa pemrograman PHP yang dikombinasikan dengan sebuah *framework agnostic* BotMan. Bot Telegram menggunakan metode webhook sebagai metode pertukaran informasi yang dilakukan secara real time. Bot Telegram diuji dengan menggunakan black box testing berbasis equivalence partitions. Hasil pengujian menunjukan bahwa bot Telegram mampu memberikan respon kepada pengguna sesuai dengan harapan sebanyak 30 dari 30 kali pengujian. Selain menguji kemampuan *bot* Telegram dalam menjalankan fungsinya, kemampuan *bot* Telegram dalam menanggapi perintah pengguna juga turut diuji. Dari 15 kali pengujian *response time*, *bot* telegram memiliki total rata – rata *response time* keseluruhan sebesar 0,9 detik yaitu termasuk ke dalam kategori cepat**.**

Kata Kunci— bot *Telegram, BotMan, Covid-19,* black box testing*,* response time, webhook

I. PENDAHULUAN

Tidak dapat dipungkiri bahwa perkembangan pesat dari teknologi informasi merupakan penyebab utama terjadinya era globalisasi datang secara lebih cepat dan tidak terduga. Perkembangan teknologi informasi seperti internet, iklan elektronik, media sosial, dan lain sebagainya berhasil menembus dinding fisik yang membatasi antar negara.[1]. Salah satu teknologi informasi yang sangat populer di kalangan masyarakat dunia adalah media sosial, setidaknya setiap orang memiliki 1 media sosial yang terpasang pada perangkat yang mereka gunakan baik *mobile* maupun desktop. [2].

Telegram merupakan salah satu media sosial yang cukup populer untuk saat ini. Sebagai aplikasi penyedia layanan pesan instan, Telegram telah dibekali dengan fitur yang lengkap seperti pesan teks, foto, video, serta berbagai macam jenis berkas dan ekstensi lainnya. Telegram juga telah mendukung pengiriman *file* dengan kapasitas besar hingga 1,5 GB [3]. Dibandingkan dengan media sosial lainnya, Telegram merupakan media sosial yang fleksibel dan mudah untuk digabungkan dengan teknologi lain. Adapun Salah satu teknologi yang sering dikombinasikan dengan Telegram adalah *bot*, yaitu teknologi cerdas yang mampu menjalankan fungsi tertentu secara otomatis [4].

Banyaknya minat masyarakat terhadap penggunaan *bot* Telegram ini, secara tidak langsung menyebabkan pesatnya perkembangan teknologi *bot* itu sendiri. Sebagai contohnya adalah integrasi *bot* Telegram dengan *application programming interface* (API) yang memungkinkan *bot* untuk melakukan pertukaran informasi dengan server maupun aplikasi lainnya [5], [6]. Pengembangan ini membuka jalan bagi *bot* Telegram untuk dimanfaatkan pada berbagai macam bidang. Khususnya dalam masa pandemi Covid-19 seperti saat ini, *bot* Telegram dapat dimanfaatkan pada bidang kesehatan.

Dalam penelitian ini, terdapat beberapa penelitian terdahulu dengan topik aplikasi *bot* Telegram pada perangkat berbasis IoT dan sistem informasi yang dijadikan peneliti sebagai referensi. Adapun beberapa referensi penelitian tersebut adalah sebagai berikut:

1. Jurnal penelitian dari Adi Krismadi dengan judul "Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan". Penelitian ini membahas tentang pengujian *black box* pada aplikasi seleksi promosi kenaikan jabatan. Hasil dari penelitian ini adalah penggunaan pengujian *black box* berbasis *equivalence partition* untuk menguji kesesuaian perangkat lunak dengan spesifikasi yang telah ditentukan serta kualitas dari aplikasi yang telah dibangun. [7].
2. Jurnal penelitian dari Rizky Parlika dengan judul " Aplikasi Mesin Penjawab Pesan Berbasis Bot Telegram, PHP, dan MySQL". Penelitian ini membahas tentang perancangan aplikasi penjawab pesan otomatis berbasis *bot* Telegram menggunakan bahasa pemrograman PHP yang terintegrasi dengan database MySQL. Hasil dari penelitian ini adalah terciptanya mesin penjawab pesan otomatis (*bot* Telegram) berbasis API menggunakan bahasa pemrograman PHP serta bagaimana struktur basis data MySQL yang digunakan dalam pembuatan *bot* [8].
3. Jurnal penelitian dari Moh. Anshori Aris Widya dengan judul "Pengembangan Telegram Bot Engine Menggunakan Metode Webhook Dalam Rangka Peningkatan Waktu Layanan E-Government". Pada penelitian ini, *bot* Telegram dibangun menggunakan metode pertukaran informasi *webhook* dengan metode penelitian *research and development* (R&D) sebagai pendekatan pengembangannya. Hasil dari penelitian ini

adalah aplikasi *bot* Telegram *engine* yang terintegrasi dengan sistem *E-Governmen* [11].

Penelitian yang dilakukan ini jika dibandingkan dengan peneliti lainnya terletak pada metode pengembangan bot Telegram yang digunakan. Pada penelitian ini, *bot* Telegram dibangun dengan menggabungkan 2 metode yang digunakan pada penelitian referensi. Peneliti membuat *bot* Telegram baru berbasis *webhook* yang dikombinasikan dengan sebuah *framework agnostic* BotMan menggunakan bahasa pemrograman PHP. *Bot* Telegram yang telah dibuat kemudian dihubungkan dengan sistem presensi RFID dan pengukuran suhu tubuh berbasis IoT dan dimanfaatkan sebagai media pemantuan data, media pelaporan dan notifikasi. Berbeda dengan penelitian referensi yang hanya digunakan untuk membalas pesan sederhana.

Pengambilan serta pemantauan data pada database MySQL pada penelitian ini dilakukan dengan menggunakan Google Spreadsheet yang telah diatur dengan *add-on* KPIBees sehingga memungkinkan bot Telegram untuk mengirimkan data dalam jumlah lebih banyak apabila dibandingkan dengan bot Telegram penelitian referensi yang hanya mampu menampung data maksimal 1000 karakter

II. BAHAN DAN METODE

Research and Development (R&D) merupakan metode yang digunakan dalam penelitian ini. Metode penelitian ini bertujuan untuk mengembangkan, menghasilkan, dan menguji sebuah produk tertentu sehingga dapat diketahui keefektifan dan kualitas dari produk. Adapun model penelitian R&D yang digunakan dalam penelitian ini adalah model Borg dan Gall (1989) sebagai berikut:

A. Penggalian Informasi dan Pengumpulan Data

Tahapan ini merupakan studi awal yang dijadikan dasar dari tahapan yang dilaksanakan selanjutnya. Adapun tahapan ini diawali dengan melakukan studi literatur, yaitu melakukan pengkajian terhadap hal - hal yang berkaitan dengan *bot* Telegram melalui artikel internet, buku, maupun jurnal penelitian. Kemudian dilanjutkan dengan studi lapangan yaitu menganalisis kebutuhan yang berkaitan dengan sistem presensi dan pengukuran suhu tubuh berbasis IoT.

B. Perencanaan

Tahapan ini merupakan tahapan perancangan dan penentuan desain – desain sistem yang dibangun dan dikembangkan. Adapun desain yang dimaksud adalah desain asitektur sistem, desain *user interface*, *entity relationship diagaram* (ERD) dan diagram basis data. Pada tahapan ini juga dilakukan pengembangan bentuk permulaan (prototipe) awal *bot* Telegram.

C. Pre-Eliminary Testing

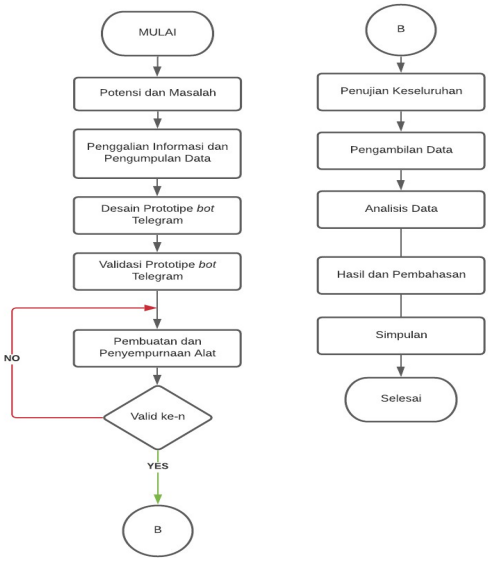
Pada tahapan ini dilakukan pengujian awal dari prototipe *bot* Telegram yang telah dibuat secara terbatas. Pengujian ini dilakukan dengan tujuan untuk mengetahui apakah prototipe *bot* Telegram yang dikembangkan mampu menjalankan fungsi – fungsinya sebelum dikembangkan secara lebih lanjut pada tahapan selanjutnya.

D. Main Product Revision

Pada tahapan ini dilakukan perbaikan atas beberapa kesalahan yang ditemukan pada tahapan sebelumnya. Tujuan dari tahapan ini adalah untuk menyempurnakan *bot* Telegram yang telah dirancang agar tidak melenceng dari rencana yang telah ditentukan.

E. Main Field Testing

Pada tahapan ini dilakukan pengujian secara menyeluruh dari bot Telegram yang telah dibuat dan memastikan bahwa bot telegram dapat berjalan dengan baik setelah diimplementasikan di lapangan. Pengujian ini dilakukan dengan menggunakan metode *black box testing* berbasis *equivalence partitions* untuk menguji kesesuaian antara perintah pengguna dengan hasil yang diberikan oleh *bot* Telegram. Pada tahapan ini juga dilakukan pengujian *response time* dari *bot* Telegram dengan tujuan untuk mengetahui kemampuan *bot* Telegram dalam memproses perintah dari pengguna.



Gambar 1. Tahapan Penelitian

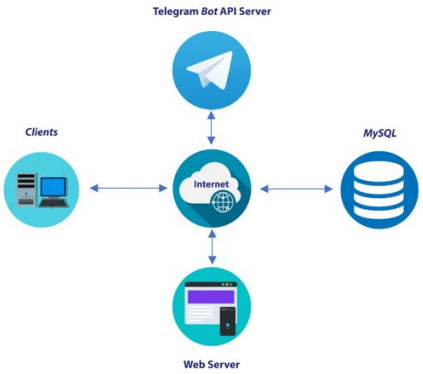
Berdasarkan Gambar 1 tersebut, penelitian ini memliki beberapa tahapan yaitu: Sedangkan tahapan penelitian yang digunakan

A. Desain Arsitektur Sistem

Tahapan awal perancangan *bot* Telegram pada penelitian ini dilakukan dengan merancang arsitektur sistem *bot* Telegram. Tahap perancangan ini penting dilakukan agar desain bot Telegram, sesuai dengan analisis kebutuhan. Selain itu, tahap ini juga berfungsi untuk memberikan gambaran pertukaran informasi dari bot Telegram. Desain arsitektur sistem bot telegram tampak pada gambar 2.

Berdasarkan desain sistem, *bot* Telegram berfungsi untuk menerima dan menjalankan perintah dari pengguna. Perintah tersebut kemudian dikirimkan oleh *bot* Telegram ke sebuah *web* server dengan menggunakan API sebagai media penghubung. *Web* server akan memproses perintah tersebut dan melakukan pengambilan data pada *database* MySQL sesuai dengan permintaan pengguna. Data permintaan tersebut selanjutnya dikirimkan kembali ke

pengguna melalui *bot* Telegram dengan menggunakan mekanisme yang sama.



Gambar 2. Desain Arsitektur *bot* telegram

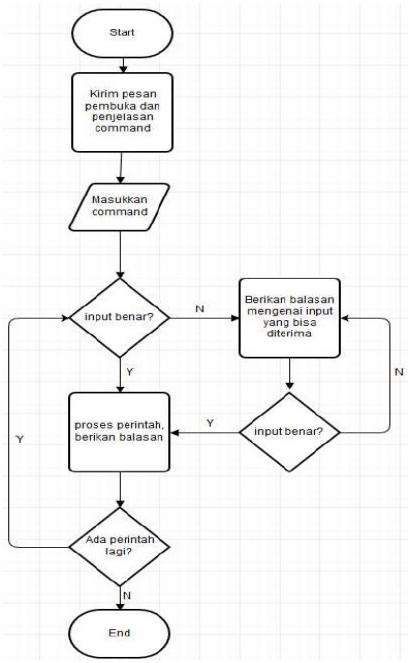
Berikut adalah daftar perintah *bot* Telegram yang terdapat pada penelitian ini:

1) /Start, perintah ini merupakan perintah pertama yang harus dikirimkan oleh pengguna apabila ingin menggunakan *bot* Telegram untuk pertama kalinya.
2) /Help, perintah ini digunakan untuk mengetahui fungsi setiap perintah yang terdapat pada *bot* telegram.
3) /Data_presensi, perintah ini digunakan untuk melihat data presensi dan pengukuran suhu tubuh pengunjung yang tersimpan.
4) /Data_covid, perintah ini digunakan untuk melihat data pelaporan kasus Covid-19.
5) /Data_penyintas, perintah ini digunakan untuk melihat data penyintas Covid-19.
6) /Pelaporan_covid, perintah ini digunakan untuk melakukan pelaporan kasus Covid-19 melalui Telegram.
7) /Pelaporan_penyintas, perintah ini digunakan untuk melakukan pelaporan penyintas Covid-19 melalui Telegram.
8) /Saran, perintah ini digunakan untuk melihat saran yang masuk pada sistem presensi dan pengukuran suhu tubuh berbasis IoT.

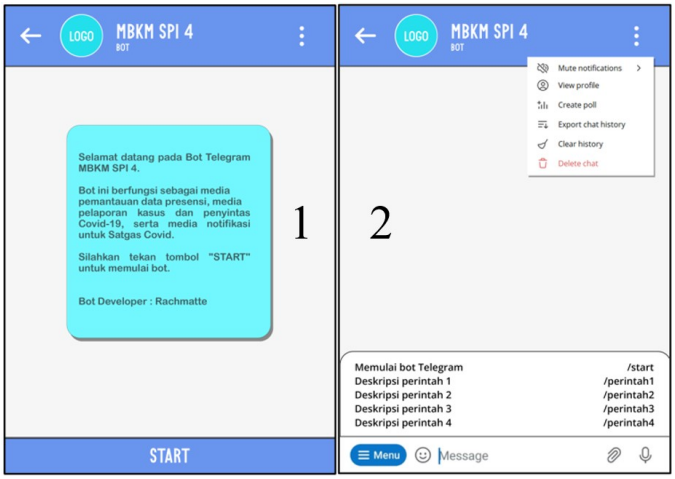
Dalam menjalankan fungsinya, *bot* Telegram hanya akan memproses perintah pengguna apabila input perintah yang diberikan sesuai dengan ketentuan sistem. Apabila input perintah yang diberikan pengguna tidak sesuai, *bot* Telegram tidak akan merespon perintah tersebut sebelum pengguna memberikan input perintah yang sesuai. Secara lebih rinci cara kerja dari bot Telegram pada penelitian ini dapat dilihat melalui diagram alir pada gambar 3.

B. *Desain User Interface*

Untuk memberikan gambaran konsep desain antarmuka *bot* Telegram, peneliti membuat sebuah *mockup* yang dapat dilihat pada gambar 4. Berdasarkan *mockup* yang telah dibuat, terdapat 2 jenis model antarmuka yang diterapkan pada *bot* Telegram. Model antarmuka 1 merupakan antarmuka awal saat *user* pertama kali menggunakan *bot* Telegram. Pada antarmuka ini terdapat sebuah deskripsi mengenai fungsi serta petunjuk penggunaan *bot* Telegram. Selain itu terdapat juga sebuah tombol "START" yang harus ditekan oleh *user* untuk memulai *bot* Telegram pertama kalinya.



Gambar 3. Diagram alir *bot* Telegram



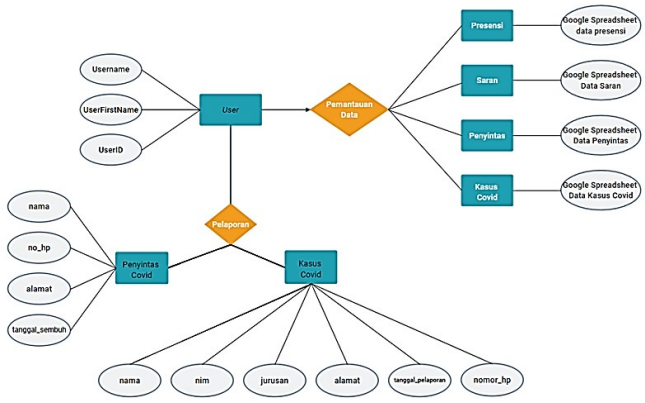
Gambar 4. *Mockup bot* Telegram

Sedangkan model antarmuka 2 merupakan antarmuka saat *user* telah menggunakan *bot* Telegram sebelumnya. Model antarmuka ini terbagi menjadi 3 bagian yaitu bagian atas yang terdiri dari tombol kembali, logo bot, nama bot, dan tombol pengaturan. Bagian tubuh untuk menampilkan perintah dan balasan dari *bot* Telegram. Serta bagian bawah yang terdiri dari menu, emotikon, pesan teks, pesan suara, dan lampiran file. Untuk memudahkan pengguna dalam penginputan perintah, pada antarmuka ini terdapat sebuah fitur khusus yaitu "menu" untuk menampilkan daftar perintah yang terdapat pada *bot* Telegram. Dengan fitur ini, pengguna hanya perlu memilih perintah yang diinginkan dan secara otomatis *bot* Telegram menjalankan perintah tersebut.

C. *Desain ER dan Basis Data*

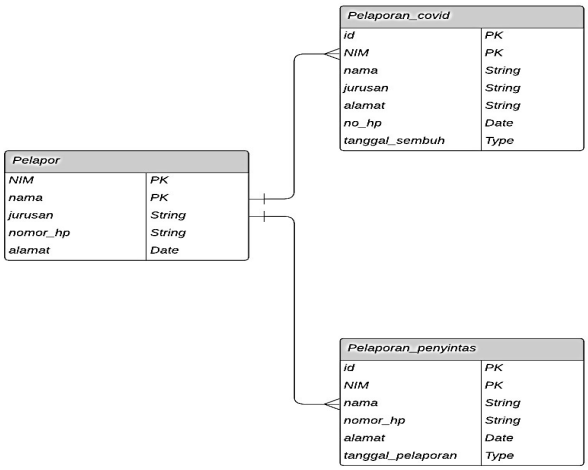
Setelah melakukan perancangan antarmuka, tahapan selanjutnya adalah membuat *entity relationship diagram* (ERD) dari *bot* Telegram. Pembuatan ERD ini berfungsi sebagai pemodelan dari basis data yang dibuat. Dengan

menggunakan ERD ini, data – data yang memiliki relasi dengan *database* dapat diketahui. Adapun berikut adalah ERD dari *bot* Telegram pada penelitian ini.



Gambar 5. Diagram ER *bot* Telegram

Berdasarkan ERD *bot* Telegram seperti gambar 5, menunjukan informasi berupa atribut, entitas, serta proses yang terjadi saat pengguna menggunakan *bot* Telegram. Dengan informasi tersebut, peneliti dapat menyusun sebuah *database* untuk kemudian diterapkan pada *bot* Telegram.



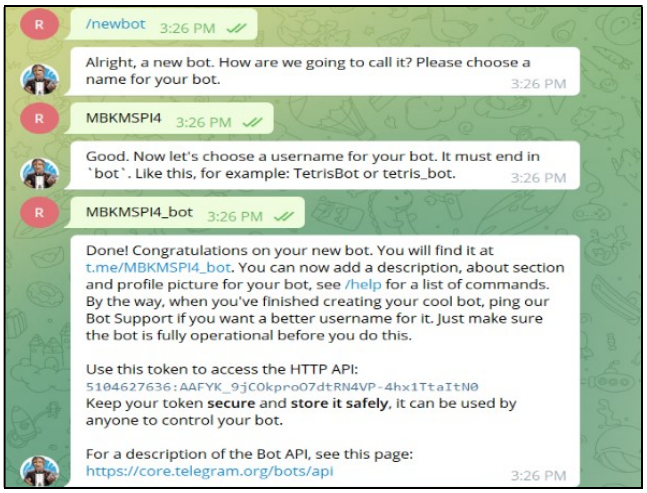
Gambar 6. Database Relational *bot* Telegram

III. HASIL DAN PEMBAHASAN

A. *Pembuatan Bot Telegram*

Pembuatan *bot* Telegram pada penelitian ini dilakukan pada aplikasi Telegram dengan memanfaatkan channel @*bot*father, yaitu sebuah channel resmi dari Telegram yang berfungsi untuk menciptakan sebuah *bot* [8], [9]. Berikut adalah tahapan pembuatan *bot* Telegram melalui channel *Bot*father:

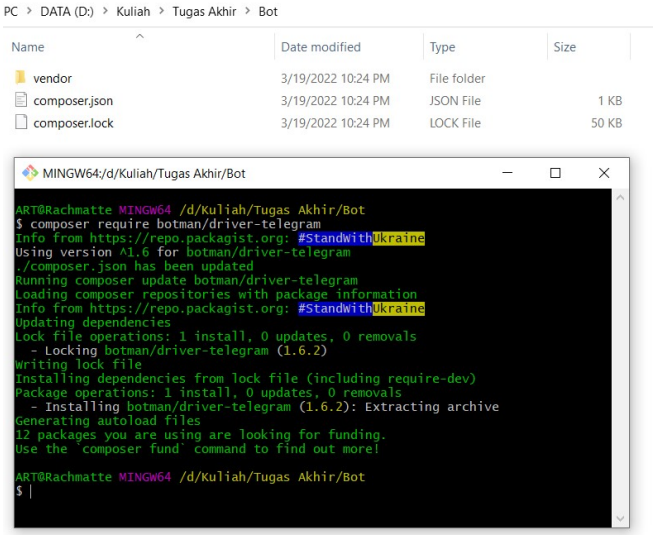
1) Mengakses channel Telegram @*bot*father.
2) Memasukan perintah /start untuk memulai *bot Bot*father.
3) Memasukan perintah /new*bot* pada *Bot*father untuk membuat *bot* baru.
4) Memberikan nama pada *bot* yang ingin dibuat.
5) *Bot* berhasil dibuat, *Bot*father mengirimkan pesan balasan berisi token HTTP API yang digunakan untuk pengembangan *bot*.



Gambar 7. Pembuatan *bot* Telegram menggunakan *Bot*father

B. *Instalasi BotMan*

*Bot*Man adalah framework agnostic berbasis PHP yang dirancang untuk mempermudah pengembangan inovasi *bot* pada berbagai macam platform perpesanan, seperti Slack, Telegram, Facebook Messenger, dan lain sebagainya. Peneliti memilih *Bot*Man sebagai library pengembangan *bot* Telegram pada penelitian ini dikarenakan proses instalasi *Bot*Man sangat mudah dan praktis. Sebagai *framework agnostic*, *Bot*man memungkinkan peneliti untuk menggunakan basis kode yang telah dibuat tanpa perlu memikirkan jenis *framework* yang digunakan [10].



Gambar 8. Instalasi *Bot*Man menggunakan *software* Git

Proses instalasi *Bot*Man seperti Gambar 8 pada penelitian ini dilakukan dengan menggunakan *software* Git, dengan tahapan instalasi sebagai berikut:

1) Membuka aplikasi GitBash pada direktori yang diinginkan.
2) Memasukkan perintah **$ composer global require *botman*/*botman*** pada terminal GitBash.
3) Memasukkan perintah **$ composer require *botman*/driver-telegram** pada terminal GitBash untuk instalasi library *Bot*man platform Telegram.

4) Apabila berhasil maka pada direktori instalasi terdapat sebuah folder vendor dan dua buah file dengan nama composer.json dan composer.lock.

C. *Pembuatan Kode Program Bot Telegram*

Kode program *bot* Telegram yang digunakan oleh tim peneliti dibuat menggunakan bahasa pemrograman PHP. Kode program tersebut terbagi kedalam 6 buah file PHP dan 1 buah file token API berekstensi text (.txt) yang dapat diunduh melalui alamat https://github.com/rachmatte/*bot*Telegram.git. Pembahasan terkait kode program *bot* Telegram yang digunakan dapat dilihat pada tabel 1.

TABEL I. PEMBAHASAN KODE PROGRAM

No	Nama File	Pembahasan
1	*Bot*.php	Kode program *bot*.php merupakan kode program utama dari *bot* Telegram yang berfungsi untuk membaca masukan perintah serta mengirimkan data atau informasi yang diinginkan oleh pengguna. Serta menjadi penghubung antara kode program satu dengan kode program lainnya sehingga dapat berjalan dalam satu kesatuan.
2	ConfigDB.php	Kode program ini berfungsi sebagai penghubung antara *bot* Telegram dengan basis data MySQL.
3	insertDataLaporan.php	Kode program ini berfungsi untuk memasukkan data yang diberikan oleh pengguna pada saat pelaporan kasus virus Covid-19 ke dalam basis data MySQL.
4	insertDataPenyintas.php	Kode program ini berfungsi untuk memasukkan data yang diberikan oleh pengguna pada saat pelaporan penyintas virus Covid-19 ke dalam basis data MySQL.
5	NotifikasiCovid.php	Kode program NotifikasiCovid.php merupakan kode program yang berfungsi untuk memberikan notifikasi apabila terdapat pelaporan kasus virus Covid-19.
6	NotifikasiPenyintas.php	Kode program NotifikasiCovid.php merupakan kode program yang berfungsi untuk memberikan notifikasi apabila terdapat pelaporan penyintas virus Covid-19.
7	Token.txt	Token.txt merupakan sebuah file yang berisi token API dari *bot* Telegram. Token API tersebut akan diakses oleh masing – masing kode program sehingga dapat terhubung dan melakukan pertukaran informasi dengan server Telegram.

D. *Pengaturan Webhook*

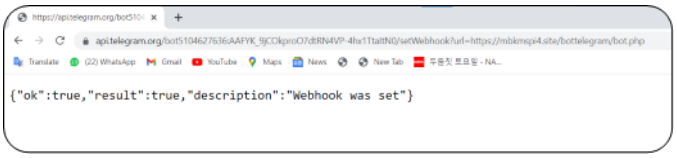
Metode pertukaran informasi *bot* Telegram yang digunakan pada penelitian ini adalah dengan menggunakan metode *webhook*. Dengan demikian, proses pertukaran informasi pada *bot* Telegram dapat dilakukan secara *real time* tanpa perlu menunggu interval waktu tertentu [11], [12].

Untuk mengatur *webhook, bot* Telegram yang telah dibuat harus terkoneksi dengan sebuah *hosting* [13]. Dalam penelitian ini peneliti menggunakan layanan *hosting* dari https://www.hostinger.co.id dengan alamat *hosting* yang digunakan yaitu https://mbkmspi4.site. Adapun Tahapan pengaturan *webhook bot* Telegram adalah sebagai berikut:

1) Memastikan bahwa *bot* Telegram sudah terkoneksi dengan layanan hosting.
2) Membuka peramban Google Chrome kemudian ketikkan URL webhook berikut.

Gambar 9. Format URL untuk pengaturan *webhook*

3) Menyesuaikan alamat URL dengan API *bot* Telegram dan hosting yang digunakan.
4) Pengaturan webhook berhasil.

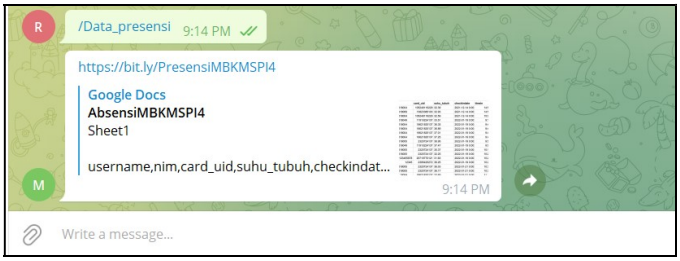


Gambar 10. Pengaturan *webhook* berhasil

E. *Penerapan Bot Telegram Sebagai Media Pemantauan Data*

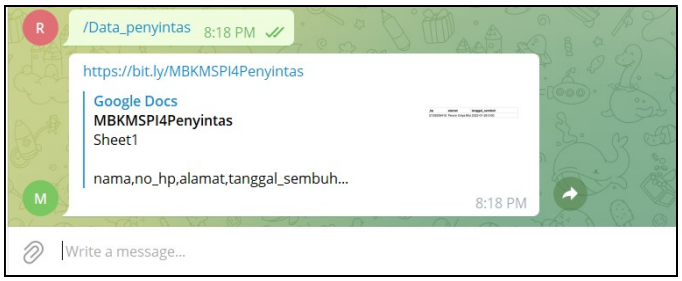
Bot Telegram yang telah dibuat dalam penelitian ini kemudian diterapkan pada salah satu alat hasil dari kegiatan MBKM Studi Independen Universitas Jenderal Soedirman yaitu "Sistem Presensi dan Pengukuran Suhu Tubuh Berbasis *Internet Of Things (IoT)"* sebagai media pelaporan, media pemberitahuan, serta sebagai media pemantauan data pada sistem.

Penerapan *bot* telegram sebagai media pemantauan yang bertujuan untuk memudahkan pemantauan data pada sistem presensi dan pengukuran suhu tubuh berbasis IoT. Dalam penelitian ini, data yang terdapat pada *database* MySQL dimuat ke dalam Google Spreadsheet dengan menggunakan *add-on* KPIBees. *Link* Google Spreadsheet inilah yang dikirimkan *bot* Telegram kepada pengguna untuk pemantauan data. Pada *bot* Telegram yang telah dibuat, terdapat 3 buah perintah yang digunakan untuk melakukan pemantauan data yaitu /Data_presensi, /Data_covid, /Data_penyintas.



Gambar 11. Pemantauan data presensi dan pengukuran suhu tubuh menggunakan *bot* Telegram

Gambar 12. Pemantauan data pelaporan kasus Covid-19 pada sistem menggunakan bot Telegram

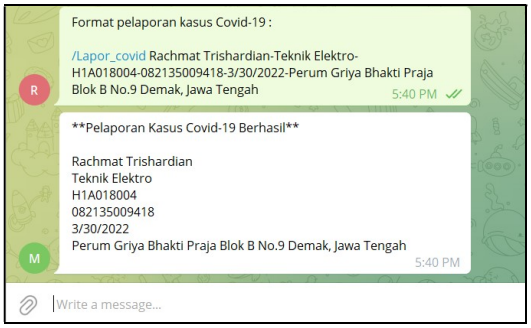


Gambar 13. Pemantauan data penyintas Covid-19 pada sistem menggunakan *bot* Telegram

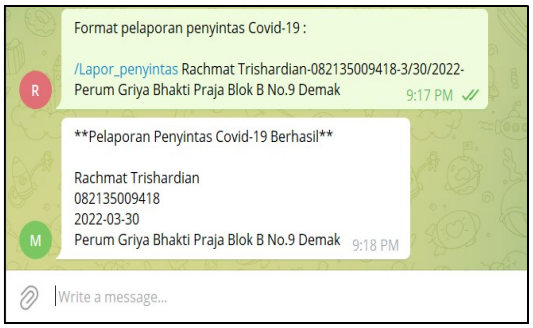
Penerapan *bot* Telegram sebagai media pemantauan dapat menjadi jalan bagi masyarakat untuk melakukan tindakan pencegahan penyebaran Covid-19. Apabila terdapat kasus baru infeksi virus Covid-19, masyarakat dapat memanfaatkan *bot* Telegram untuk mempermudah pelacakan kontak erat virus Covid-19 melalui data presensi dan pengukuran suhu tubuh dari sistem. Masyarakat juga dapat melihat siapa saja pasien Covid-19 yang sedang berada dalam tahap penyembuhan dan membutuhkan bantuan dari data pelaporan. Semakin cepat tindakan pencegahan penyebaran virus Covid-19 dilakukan maka harapanya rantai penyebaran virus Covid-19 semakin cepat untuk terputus.

F. Penerapan Bot Telegram Sebagai Media Pelaporan

Penerapan *bot* Telegram sebagai media pelaporan pada sistem presensi dan pengukuran suhu tubuh berbasis IoT bertujuan untuk mempermudah masyarakat dalam melakukan pelaporan kasus baru virus Covid-19 maupun pelaporan penyintas virus Covid-19. *Bot* Telegram ini menjadi alternatif lain untuk melakukan pelaporan selain melalui situs web dari sistem. Dalam penelitian ini, terdapat 2 jenis pelaporan yang dapat dilakukan oleh *Bot* Telegram yaitu pelaporan kasus virus Covid-19 dan pelaporan penyintas virus Covid-19.



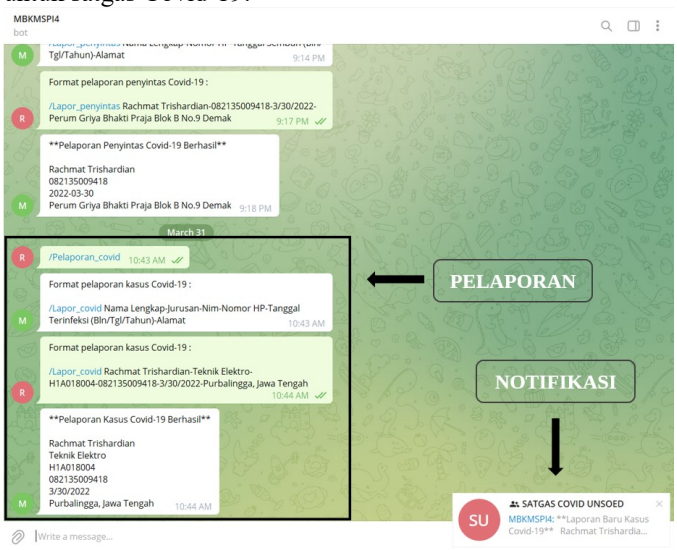
Gambar 14. Pelaporan kasus virus Covid-19 melalui *bot* Telegram



Gambar 15. Pelaporan penyintas Covid-19 melalui *bot* Telegram

G. Penerapan Bot Telegram Sebagai Media Pemberitahuan

Selain sebagai media pelaporan, *bot* Telegram pada penelitian ini juga berfungsi sebagai media pemberitahuan atau notifikasi [14]. Apabila ditemukan laporan kasus Covid-19 serta penyintasnya pada sistem presensi dan pengukuran suhu tubuh, *bot* Telegram secara otomatis mengirimkan pesan pemberitahuan ke sebuah grup khusus untuk satgas Covid-19.



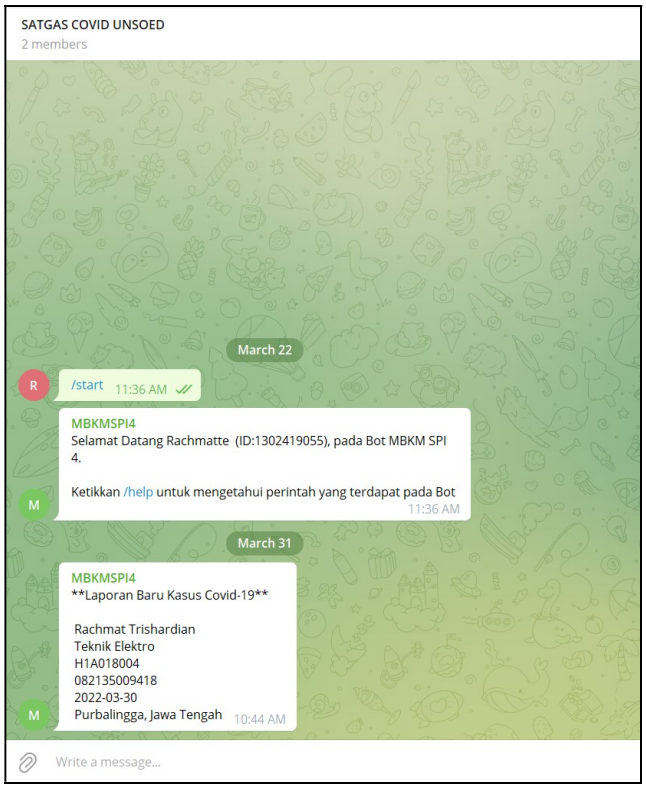
Gambar 16. Pemberitahuan pelaporan kasus Covid-19 pada grup satgas covid

Penerapan *bot* Telegram sebagai media pemberitahuan [15] bertujuan untuk memastikan pelaporan kasus Covid-19 maupun penyintas Covid-19 dapat secara langsung diketahui oleh satgas Covid-19 sehingga tindakan pencegahan virus Covid-19 dapat segera dilakukan.

H. Pengujian Black Box Bot Telegram.

Pada penelitian ini, *bot* Telegram yang telah dibuat dan diterapkan pada sistem presensi dan pengukuran suhu tubuh berbasis IoT kemudian diuji dengan menggunakan *black box testing*. Pengujian *black box testing* bertujuan untuk menguji fungsionalitas dari *bot* Telegram, apakah input dan *output* dari *bot* Telegram yang dibuat sudah sesuai dengan apa yang diharapkan oleh pengembang [16], [17].

Salah satu cara untuk melakukan pengujian *black box testing* adalah menggunakan *equivalence partitions,* yaitu dengan membagi domain input dari *bot* Telegram menjadi beberapa partisi. Berdasarkan partisi tersebut, maka dapat ditentukan sebuah *test case* yang digunakan sebagai acuan dalam melakukan pengujian [18], [19].



Gambar 17. Pesan Pemberitahuan pelaporan kasus Covid-19 pada grup satgas covid Unsoed

Hasil pengujian dari *test case* kemudian digunakan untuk menyimpulkan [20] apakah *bot* Telegram sudah berfungsi dengan baik sesuai dengan harapan atau terdapat masalah yang perlu diperbaiki. Pengujian yang telah dilakukan menggunakan tiga kriteria pengujian yaitu Pengujian Bot Telegram Pada Telegram Mobile, Pengujian Bot Telegram pada Telegram Desktop, dan Pengujian Bot Telegram Pada Telegram Web. Pengujian yang dilakukan baik secara secara simultan maupun secara mandiri. Hasil pengujian yang diperoleh ditampilkan pada Tabel II, Gambar 18, Gambar 19 dan Gambar 20 secara berurutan.

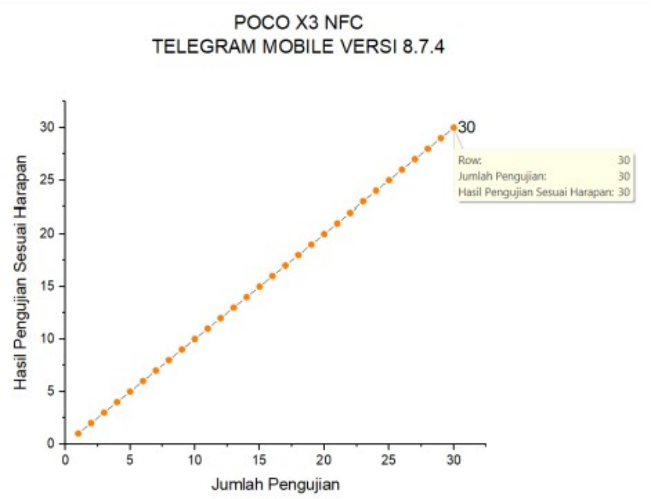
Berdasarkan pengujian yang dilakukan, dapat diketahui bahwa *bot* Telegram dapat berfungsi dengan baik. Hal ini dibuktikan dengan kesesuaian antara hasil pengujian dengan hasil yang diharapkan pada *test case* yang telah ditentukan.

TABEL II. PENGUJIAN *BOT* TELEGRAM PADA TELEGRAM MOBILE

No	***Device***	**Deskripsi pengujian**	**Harapan**	**Jumlah Pengujian**	**Hasil**
1	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Menjalankan perintah "/start" pada *bot* Telegram.	Ucapan selamat datang dan petunjuk penggunaan *bot* Telegram.	30	Sesuai harapan
2	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Menjalankan perintah "/help" pada *bot* Telegram	Pesan balasan berupa daftar perintah yang terdapat pada *bot* Telegram.	30	Sesuai harapan
3	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Menjalankan perintah "/Data_presensi" pada *bot* Telegram	Pesan balasan berupa *link* Google Spreadsheet data presensi dan pengukuran suhu tubuh.	30	Sesuai harapan
4	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Menjalankan perintah "/Data_covid" pada *bot* Telegram	Pesan balasan berupa *link* Google Spreadsheet data pelaporan kasus Covid-19.	30	Sesuai harapan
5	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Menjalankan perintah "/Data_penyintas" pada *bot* Telegram	Pesan balasan berupa *link* Google Spreadsheet data pelaporan penyintas Covid-19.	30	Sesuai harapan
6	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Menjalankan perintah "/Saran" pada *bot* Telegram	Pesan balasan berupa *link* Google Spreadsheet data saran yang masuk pada sistem.	30	Sesuai harapan
7	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Menjalankan perintah "/Pelaporan_covid" pada *bot* Telegram	Pesan balasan berupa format pelaporan kasus Covid-19.	30	Sesuai harapan
8	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Mengisi format pelaporan kasus covid-19 dengan data yang benar.	*Bot* Telegram menyimpan data pelaporan kasus Covid-19 ke dalam *database* MySQL dan mengirimkan notifikasi pelaporan kasus Covid-19	30	Sesuai harapan
9	POCO X3 NFC (Telegram Mobile Versi	Mengisi format pelaporan kasus covid-19 dengan data yang salah.	*Bot* Telegram menolak menyimpan data pelaporan kasus Covid-19	30	Sesuai harapan

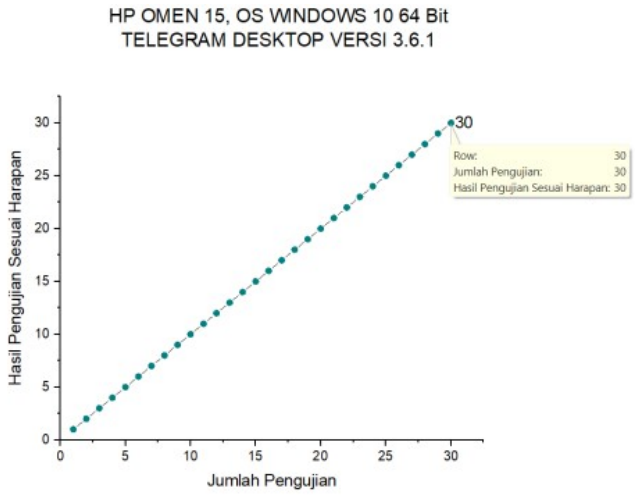
No	*Device*	Deskripsi pengujian	Harapan	Jumlah Pengujian	Hasil
	8.7.4)				
10	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Menjalankan perintah "/Pelaporan_penyintas" pada *bot* Telegram	Pesan balasan berupa format pelaporan penyintas Covid-19.	30	Sesuai harapan
11	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Mengisi format pelaporan kasus covid-19 dengan data yang benar.	*Bot* Telegram menyimpan data pelaporan penyintas ke dalam *database* MySQL dan mengirimkan notifikasi pelaporan penyintas Covid-19	30	Sesuai harapan
12	POCO X3 NFC (Telegram Mobile Versi 8.7.4)	Mengisi format pelaporan kasus covid-19 dengan data yang salah.	*Bot* Telegram menolak menyimpan data pelaporan penyintas Covid-19	30	Sesuai harapan

Gambar 18, Gambar 19, dan Gambar 20 secara berurutan menunjukan grafik perbandingan antara jumlah pengujian dengan hasil pengujian sesuai harapan pada Telegram *mobile,* Telegram deskop, dan telegram versi web.

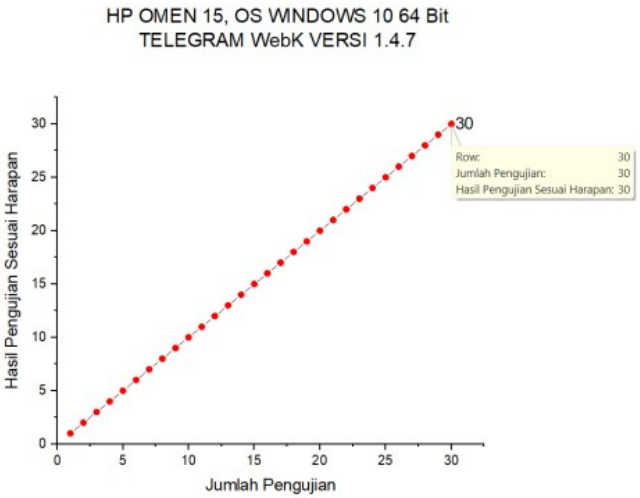


Gambar 18. Grafik pengujian *black box bot* Telegram

Berdasarkan pengujian *test case* sebanyak 30 kali, bot Telegram mampu memberikan hasil sesuai harapan sebanyak 30 yang secara rinci dapat dilihat tabel II. Hasil pengujian *test case* menunjukan tidak terdapat *bug* maupun *defect* yang ditemukan pada *bot* telegram dalam menjalankan fungsinya sebagai media pemantauan, media pemberitahuan, dan media pelaporan pada ketiga versi telegram tersebut.



Gambar 19. Grafik pengujian *black box bot* Telegram



Gambar 20. Grafik pengujian *black box bot* Telegram

A. Pengujian *Response Time Bot* Telegram.

Pengujian *response time bot* Telegram bertujuan untuk mengukur kemampuan *bot* Telegram dimana dalam hal ini adalah waktu yang dibutuhkan *bot* Telegram untuk menanggapi perintah dari pengguna. Adapun untuk mengukur *response time bot* Telegram peneliti menggunakan kode program PHP berikut.

```
$waktu = microtime(true) - $_SERVER["REQUEST_TIME_FLOAT"];
```

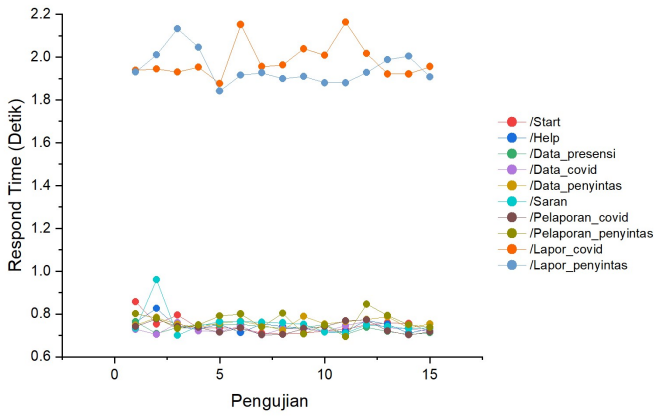
Gambar 21. Kode program untuk mengukur *response time*

Pada kode program tersebut terdapat dua buah fungsi yang digunakan yaitu **$_SERVER"REQUEST_TIME_FLOAT"** untuk mengukur waktu yang dibutuhkan server web saat melakukan penyerahan pemrosesan ke PHP dan fungsi **microtime(true)** untuk mengukur lama waktu pemrosesan pada potongan kode program yang dispesifikasikan. Hasil

pengukuran response time pada fungsi **microtime(true)** kemudian dikurangi dengan hasil pengukuran *response time* fungsi **$_SERVER"REQUEST_TIME_FLOAT"** sehingga dapat diketahui secara spesifik *response time* bersih kode program dalam menanggapi perintah pengguna. Berikut adalah hasil pengujian *response time* yang dilakukan oleh peneliti dengan menggunakan jaringan Indihome dengan kecepatan 10 mbps.

TABEL III Respon Time Bot Telegram

No	**Jaringan**	**Fungsi**	**Jumlah Pengujian**	**Rata – Rata Response Time**
1	Indihome 10 Mbps	/start.	15	0.745
2	Indihome 10 Mbps	/help	15	0.743
3	Indihome 10 Mbps	/Data_presensi	15	0.733
4	Indihome 10 Mbps	/Data_covid	15	0.731
5	Indihome 10 Mbps	/Data_penyintas	15	0.758
6	Indihome 10 Mbps	/Saran	15	0.755
7	Indihome 10 Mbps	/Pelaporan_covid	15	0.734
8	Indihome 10 Mbps	/Pelaporan_penyintas	15	0.765
9	Indihome 10 Mbps	/Lapor_covid	15	1.983
10	Indihome 10 Mbps	/Lapor_penyintas	15	1.947
Rata – rata *response time* keseluruhan				**0.989**



Gambar 22. Grafik pengujian *response time bot* Telegram

Berdasarkan hasil pengujian response time *bot* Telegram sebanyak 15 kali, dapat diketahui response time rata – rata dari bot Telegram dalam menanggapi perintah pengguna adalah sebesar 0.989 detik. Secara lebih rinci response time untuk masing – masing perintah dari *bot* Telegram dapat dilihat pada tabel 9.

Berdasarkan hasil pengujian tersebut, dapat diketahui bahwa perintah *bot* Telegram /Lapor_covid dan /Lapor_penyintas memiliki rata – rata *response time* yang lebih lama, jika dibandingkan dengan perintah lainya yang hanya memerlukan rata – rata *response time* sebesar 0,7 detik. Rata-rata kecepatan respon kedua perinta tersebut selama 1,9 detik. Hal tersebut disebabkan karena kedua perintah tersebut digunakan untuk melakukan pelaporan kasus dan penyintas Covid-19 sehingga membutuhkan waktu yang lebih lama karena membutuhkan akses koneksi terhadap *database* MySQL. Direkomendasikan untuk mengimplementasikan bot Telegram ini dengan layanan hosting dan provider penyedia koneksi internet yang handal pada lingkungan operasionalnya untuk mengejar kepuasan pengguna dengan terjaminnya tingkat ketersediaan layanan diseminasi ini.

IV. KESIMPULAN

Dalam penelitian ini, *bot* Telegram berfungsi sebagai media pemantauan data, media pelaporan, dan media pemberitahuan pada sistem presensi dan pengukuran suhu tubuh berbasis IoT. Penggunaan metode *webhook* sebagai proses pertukaran informasi *bot* Telegram cocok untuk digunakan pada sistem yang membutuhkan proses pertukaran data secara *real time*. Adapun penggunaan *framework agnostic BotMan* sangat mempermudah *developer* dalam melakukan pengembangan *bot* Telegram karena basis kode yang telah dibuat dapat diterapkan dengan *BotMan* tanpa perlu menyesuaikan jenis *framework* yang digunakan.

Berdasarkan hasil pengujian *black box testing* yang telah dilakukan, dapat disimpulkan bahwa semua fitur yang terdapat pada *bot* Telegram dapat berjalan sesuai dengan harapan serta tidak terdapat temuan *bug* maupun *defect* dalam menjalankan fungsinya. Hasil pengujian *response time* menunjukan *response time* rata – rata dari *bot* Telegram dalam menanggapi perintah pengguna adalah sebesar 0.989 detik dimana waktu tersebut termasuk ke dalam kategori cepat.

Adapun beberapa saran yang peneliti berikan untuk penelitian selanjutnya yaitu, menambahkan grafik data pelaporan kasus Covid-19 dan penyintas Covid-19 yang dapat diakses oleh pengguna melalui *bot* Telegram, menambahkan fitur *bot* Telegram untuk mengubah dan mencari data yang terdapat pada sistem presensi dan pengukuran suhu tubuh berbasis IoT serta mengoptimalkan penggunaan *add-on* KPIBees pada Google Spreadsheet sehingga penyegaran permintaan data dapat dilakukan secara otomatis.

.

DAFTAR PUSTAKA

[1] R. Komalasari, "MANFAAT TEKNOLOGI INFORMASI DAN KOMUNIKASI DI MASA PANDEMI COVID 19," *tematik*, vol. 7, no. 1, pp. 38–50, Jun. 2020, doi: 10.38204/tematik.v7i1.369.

[2] L. D. Warpindyastuti and M. E. S. Sulistyawati, "Pemanfaatan Teknologi Internet Menggunakan Media Sosial Sebagai Sarana Penyebaran Informasi dan Promosi Pada MIN 18 Jakarta," *Widya Cipta*, vol. II, no. 1, pp. 91–95, 2018.

[3] F. Fitriansyah, "Penggunaan Telegram Sebagai Media Komunikasi Dalam Pembelajaran Online," *Jurnal Humaniora Bina Sarana Informatika*, vol. 20, no. 2, pp. 111–117, Sep. 2020, doi: 10.31294/jc.v20i2.

[4] H. Hasyim and R. R. P. Suroso, “Peranan Teknologi Informasi Dalam Upaya Pencegahan Virus COVID-19 di Lingkungan Universitas,” *Jurnal Ilmiah Pendidikan Teknik Elektro*, vol. 4, no. 2, pp. 124–129, Aug. 2020, doi: 10.22373/crc.v4i2.6496.

[5] Y. P. Atmojo, “*Bot* Alert Snort dengan Telegram *Bot* API pada Intrusion Detection System: Studi Kasus IDS pada Server Web,” *STMIK Pontianak*, pp. 176–180, Jul. 2018.

[6] M. Syani dan B. Saputro, “Implementasi Remote Monitoring Pada Virtual Private Server Berbasis Telegram *Bot* Api (Studi Kasus Politeknik Tedc Bandung,” siskom-kb, vol. 4, no. 2, pp. 94–111, Mar. 2021, doi: 10.47970/siskom-kb.v4i2.190.

[7] A. Krismadi, A. F. Lestari, A. Pitriyah, I. W. P. A. Mardangga, M. Astuti, and A. Saifudin, “Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan,” *JTSIA*, vol. 2, no. 4, pp. 155–161, Oct. 2019, doi: 10.32493/jtsi.v2i4.3771.

[8] R. Parlika dan A. Pratama, “APLIKASI MESIN PENJAWAB PESAN BERBASIS *BOT* TELEGRAM, PHP, DAN MYSQL,” *Universitas Pembangunan Nasional “Veteran”Jawa Timur*, vol. 14, no. 3, pp. 1–9, Oct. 2019, doi: 10.33005/scan.v14i3.1624.

[9] R. Wijayanto, F. Pradana, dan F. A. Bachtiar, “Pembangunan Sistem Chat*bot* Informasi Objek Wisata Kota Malang berbasis Web,” vol. 4, no. 5, pp. 1524–1530, May 2020.

[10] A. D. Kusuma, “PENGGUNAAN TELEGRAM *BOT* PADA TELEGRAM MESSENGER DENGAN METODE WEBHOOKS UNTUK SISTEM PEMINJAMAN INFRASTRUKTUR DI UIN MAULANA MALIK IBRAHIM MALANG,” UIN MAULANA MALIK IBRAHIM MALANG, Malang, 2018.

[11] Moh. A. Aris Widya dan P. Airlangga, “Pengembangan Telegram *Bot* Engine Menggunakan Metode Webhook Dalam Rangka Peningkatan Waktu Layanan E-Government,” *SAINTEKBU*, vol. 12, no. 2, pp. 13–22, Oct. 2020, doi: 10.32764/saintekbu.v12i2.884.

[12] R. Parlika dan A. Pratama, “APLIKASI PENAMPIL BERITA REALTIME BERBASIS *BOT* TELEGRAM MENGGUNAKAN API WEB (APBR VERSI 1.0),” pp. 17-20, Sep. 2019.

[13] F. Panjaitan dan R. Syafari, “PEMANFAATAN NOTIFIKASI TELEGRAM UNTUK MONITORING JARINGAN,” vol. 10, no. 2, pp. 725-732, Nov. 2019.

[14] A. Krismadi, A. F. Lestari, A. Pitriyah, I. W. P. A. Mardangga, M. Astuti, and A. Saifudin, “Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan,” *JTSIA*, vol. 2, no. 4, pp. 155–161, Oct. 2019, doi: 10.32493/jtsi.v2i4.3771.

[15] W. N. Cholifah, Y. Yulianingsih, dan S. M. Sagita, “Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap,” *STRING*, vol. 3, no. 2, pp. 206-210, Dec. 2018, doi: 10.30998/string.v3i2.3048.

[16] A. Utomo, Y. Sutanto, E. Tiningrum, dan E. M. Susilowati, “PENGUJIAN APLIKASI TRANSAKSI PERDAGANGAN MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS,” *JBT*, vol. 4, no. 2, pp. 133–140, Dec. 2020, doi: 10.24123/jbt.v4i2.2170.

[17] Y. D. Wijaya and M. W. Astuti, “PENGUJIAN BLACKBOX SISTEM INFORMASI PENILAIAN KINERJA KARYAWAN PT INKA (PERSERO) BERBASIS EQUIVALENCE PARTITIONS,” *JDTI*, vol. 4, no. 1, pp. 22–26, Mar. 2021, doi: 10.32502/digital.v4i1.3163.

[18] A. Agustian, I. Andryani, S. Khoerunisa, A. Pangestu, and A. Saifudin, “Implementasi Teknik Equivalence Partitioning pada Pengujian Aplikasi E-learning Berbasis Web,” *JTSIA*, vol. 3, no. 3, pp. 178–184, Aug. 2020, doi: 10.32493/jtsi.v3i3.5371.

[19] A. Rosalina, A. A. G. R. Rassi, G. Y. Hadi, R. Ubaidillah, and T. Desyani, “Pengujian Black Box pada Sistem Informasi Penjualan HI Shoe Store Menggunakan Teknik Equivalence Partitions,” *JIUP*, vol. 5, no. 1, pp. 26–29, Mar. 2020, doi: 10.32493/informatika.v5i1.3964.

[20] A. D. Mulyanto, “Pemanfaatan *Bot* Telegram Untuk Media Informasi Penelitian,” *M*, vol. 12, no. 1, pp. 49–54, Apr. 2020, doi: 10.18860/mat.v12i1.8847.