

Auto Response Messages pada Telegram Bot untuk Pelayanan Sistem Informasi Praktek Industri dan Skripsi dengan Metode *Webhook*

Aji Ridwan Syah¹, Agus Prihanto²

^{1,2} Jurusan Teknik Informatika, Prodi Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹aji.18056@mhs.unesa.ac.id

²agusprihanto@unesa.ac.id

Abstrak— Praktik industri (PI) dan Skripsi merupakan salah satu mata kuliah yang dijalani oleh mahasiswa Teknik Informatika Unesa. PI dan skripsi memiliki beberapa dokumen yang perlu dibuat dan didaftarkan agar mata kuliah bisa diselesaikan, namun saat ini tidak adanya media utama yang memberikan informasi ataupun dokumen yang berkaitan dengan mata kuliah tersebut membuat beberapa mahasiswa sedikit kesulitan untuk mencari informasi terkait. Biasanya mahasiswa mencari informasi terkait PI dan skripsi melalui kakak tingkat yang sudah pernah melaksanakan mata kuliah tersebut. Penelitian ini bertujuan untuk membangun sebuah sistem pesan otomatis (auto response messages) menggunakan telegram bot yang akan dijadikan sebagai aplikasi pesan utama dengan menggunakan metode webhook yang berfungsi sebagai penyambung antara server telegram dan server bot yang nantinya akan membuat bot mampu berjalan secara otomatis dalam waktu 24 jam penuh. File dokumen pendukung PI dan skripsi dijadikan satu oleh peneliti dalam sebuah folder pada google drive yang nantinya akan diambil link nya untuk dimasukkan kedalam kode untuk dipanggil saat pengguna memilih menu mata kuliah yang dibuat. Lalu penggunaan model seperti classification dan similarity word juga akan diadaptasikan terhadap bot. Hasil pengujian menunjukkan bahwa pengaplikasian sistem pesan otomatis (auto response messages) melalui bot telegram sudah berhasil dibuat menggunakan metode webhook dan hasil simulasi menunjukkan bahwa bot yang telah dibuat mampu memberikan respon secara otomatis dalam mencari dokumen terkait mata kuliah praktik industri dan skripsi dikarenakan metode webhook yang sudah berhasil dicek dan diset pada server. File dokumen yang dibutuhkan sudah diorganisir menjadi beberapa folder yang sudah diberi nama sesuai pada google drive. Bot juga mampu digunakan oleh beberapa perangkat sekaligus tanpa mengalami delay ataupun crash pada server. Model yang digunakan seperti classification dan similarity word pun mampu diterapkan dengan baik pada bot.

Kata Kunci — *Praktik Industri, Skripsi, Telegram, Bot, Webhook.*

I. PENDAHULUAN

Perkembangan teknologi dan Informasi, khususnya yang terjadi di Indonesia terjadi sangat dinamis. Perkembangan tersebut tentu saja berdampak pada segala bidang seperti, ekonomi, kesehatan, sosial dan tentunya pada bidang pendidikan. Dalam hal ini, Ahmad D. Marimba mengartikan

jika pendidikan merupakan bimbingan yang dilakukan secara sadar oleh pendidik kepada peserta didik dengan tujuan untuk membentuk kepribadian secara jasmani dan rohani. Dari pernyataan diatas dapat disimpulkan bahwa tujuan pendidikan bisa dicapai apabila proses pembelajaran dilakukan secara efektif serta efisien untuk dapat mencapai tujuan yang optimal [1]. Hal ini pun yang menjadi pertimbangan peneliti untuk melakukan pengembangan teknologi untuk membantu mahasiswa dalam pelaksanaan mata kuliah praktik industri dan juga Skripsi.

Praktik industri (PI) dan Skripsi merupakan salah satu mata kuliah yang dijalani oleh mahasiswa Teknik Informatika Unesa. PI dan Skripsi memiliki beberapa dokumen yang perlu dibuat dan didaftarkan agar mata kuliah bisa diselesaikan. Namun saat ini tidak adanya media utama yang memberikan informasi ataupun dokumen yang berkaitan dengan mata kuliah tersebut membuat beberapa mahasiswa sedikit kesulitan untuk mencari informasi terkait. Biasanya mahasiswa mencari informasi terkait PI dan skripsi melalui kakak tingkat yang biasanya sudah pernah melaksanakan mata kuliah tersebut. Namun terkadang ada beberapa kakak tingkat yang sudah tidak memiliki file tersebut yang akhirnya membuat mahasiswa harus mencari lagi ke kakak tingkat yang selanjutnya sampai menemukan file yang dimaksud. Namun tidak sedikit perubahan yang ada pada dokumen tersebut jadi otomatis mahasiswa harus mencari dokumen yang terbaru dari mahasiswa lain yang telah selesai melaksanakan mata kuliah tersebut. Masalah ini pun sedang dirasakan oleh peneliti yang akhirnya memiliki ide untuk membuat sebuah chatbot yang bisa digunakan untuk mencari dokumen – dokumen tentang mata kuliah PI dan Skripsi. Maka dari itu peneliti memutuskan untuk menggunakan telegram sebagai salah satu media pembuat chatbot yang sederhana dan bisa digunakan secara efisien oleh mahasiswa.

Telegram didirikan pada tahun 2013 oleh dua orang bersaudara bernama Nikolai dan Pavel Durov yang bertujuan untuk menyediakan aplikasi instant message yang dapat digunakan berkinir pesan dengan aman bagi pengguna yang tidak mengerti teknologi. Telegram memungkinkan pengguna

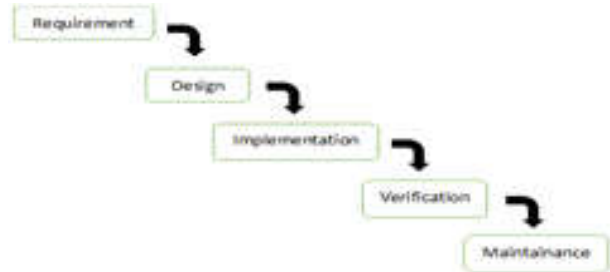
untuk mengirim pesan berupa teks, pesan suara, dan berkomunikasi dalam grup [3]. Menurut referensi [4] Penelitian sebelumnya yang dilakukan oleh Heri Khariono et al., 2021 “PEMANFAATAN BOT TELEGRAM SEBAGAI E-LEARNING UJIAN BERBASIS FILE” telegram dimanfaatkan sebagai aplikasi untuk mendukung sistem ujian berbasis file. Telegram sendiri merupakan media informasi aplikasi perpesanan yang menawarkan berbagai keunggulan yaitu fitur bot. Bot merupakan agen yang cukup interaktif, wong yang diprogram untuk pemodelan pintar [5]. Telegram sebagai salah satu layanan pesan instan merupakan sebuah aplikasi cloud base dan alat enkripsi. Telegram menyediakan enkripsi end-to-end, self destruction Messages, dan infrastruktur multi-data center. Selain itu telegram juga menyediakan wadah bagi pengembang yang ingin memanfaatkan Open API dan Protocol yang disediakan melalui pengembangan bot telegram yang didokumentasikan pada web resminya. Bot itu bisa diartikan sebagai program yang bekerja secara otomatis. Pengguna dapat berinteraksi dengan bot dengan mengirimkan pesan perintah (commands) melalui pesan private maupun group [6].

Webhook merupakan salah satu metode komunikasi informasi dalam Bot Engine Telegram. Secara keseluruhan Bot Engine telegram sendiri memiliki 2 metode komunikasi yaitu Long Pooling dan Webhook. Dibandingkan dengan Long pooling. Webhook memiliki mekanisme komunikasi yang lebih singkat sehingga lebih cepat untuk digunakan dalam merespon maupun memproses informasi sebagainya [2]. Webhook adalah konsep API yang juga disebut webcallback, HTTPpushAPI, atau reverseAPI. Metode ini akan melakukan callback secara real time dari line , ke serverwebhook yang sudah dibuat. ServerWebhook diberikan script untuk menjalankan beberapa perintah yang akan diprosesnya nanti [7].

Dengan mempertimbangkan permasalahan yang ada karena tidak adanya media utama bagi mahasiswa untuk mencari dokumen – dokumen mata kuliah praktik industri dan skripsi. Maka dari itu, peneliti mengusulkan penelitian berjudul “ Auto Response Message Pada Telegram Bot untuk Pelayanan Sistem Informasi Praktek Industri dan Skripsi Dengan Metode Webhook” dengan harapan dapat dimaksimalkan secara efektif dan efisien oleh mahasiswa agar tidak perlu kesulitan lagi dalam mencari dokumen – dokumen yang dibutuhkan. Lalu pengujian yang akan dilakukan ialah melakukan pengujian ke beberapa pengguna untuk menilai seberapa responsif telegram bot yang sudah disambungkan menggunakan metode webhook. Setelah pengujian dilakukan maka bisa dianalisa seberapa efektif penggunaan telegram bot agar kedepannya bisa menjadi sistem informasi utama di jurusan.

II. METODE PENELITIAN

Penelitian ini menggunakan metode *waterfall*. Metode *waterfall* merupakan pengembangan perangkat lunak dengan penekanan pada progresi logis yang diambil oleh *software development life cycle* (SDLC). Untuk model pengembangannya dapat dianalogikan seperti air terjun, dimana setiap tahap dikerjakan secara berurutan mulai dari atas ke. Berikut ini merupakan skema penelitian berdasarkan metode *waterfall*.



Gambar 1. Alur penelitian

1. Requirement

a. Analisis masalah

Praktik Industri dan skripsi merupakan salah satu mata kuliah wajib yang harus ditempuh mahasiswa jurusan Teknik Informatika Unesa agar bisa lulus. Praktik industri dan juga skripsi punya sistem dimana mahasiswa harus mendaftar terlebih dahulu ke tata usaha jurusan agar diperbolehkan untuk pelaksanaan mata kuliah tersebut. Dalam mendaftar mahasiswa memerlukan dokumen pendaftaran berupa formulir pendaftaran, proposal, laporan, dll. Begitu pula setelah pelaksanaan praktik industri dan skripsi adapun dokumen – dokumen yang perlu diselesaikan untuk perihal penilaian seperti laporan penilaian industri, penguji ataupun pembimbing, dll.

Masalah yang terjadi disini ialah tidak adanya media utama bagi mahasiswa dalam mencari dokumen – dokumen tersebut, cara yang biasanya dilakukan ialah meminta template dari kakak tingkat yang dirasa sudah menyelesaikan mata kuliah tersebut. Namun beberapa kakak tingkat sudah tidak memiliki file tersebut yang akhirnya harus mencari orang lain lagi untuk mencari dokumen tersebut dan biasanya meminta ke teman yang sekiranya sudah menyelesaikannya. Hal ini dirasa cukup menyusahkan apalagi dokumen yang dibutuhkan harus segera ada untuk diedit dan dikumpulkan. Maka dari itu peneliti membuat solusi dengan membuat telegram bot untuk menyelesaikan permasalahan diatas.

b. Analisis kebutuhan Data

Data yang digunakan pada penelitian ini ialah dokumen – dokumen yang menjadi salah satu syarat dari mata kuliah yang ada di jurusan Teknik informatika yaitu praktik industri dan skripsi. Berikut ini keterangan dari mata kuliah tersebut.

1). Praktik Industri

Dokumen yang digunakan untuk mengisi kebutuhan mata kuliah praktik industri terdiri dari 2 macam, yaitu pra praktik industri dan pasca praktik industri. Dokumen praktik industri ini diperoleh melalui data yang dimiliki peneliti setelah selesai melaksanakan praktik industri.

2). Skripsi

Dokumen yang digunakan untuk mengisi kebutuhan mata kuliah skripsi terdiri dari 4 macam yaitu, pra sempro, pasca sempro, pra semhas dan pasca semhas. Dokumen skripsi ini didapatkan dari dokumen – dokumen milik kakak tingkat dan juga teman senangkatan peneliti yang sudah menyelesaikan skripsi.

c. Kebutuhan perangkat

Spesifikasi dan keterangan perangkat keras maupun lunak yang digunakan sebagai perangkat pendukung dalam penelitian ini adalah sebagai berikut :

Tabel 1. Kebutuhan alat

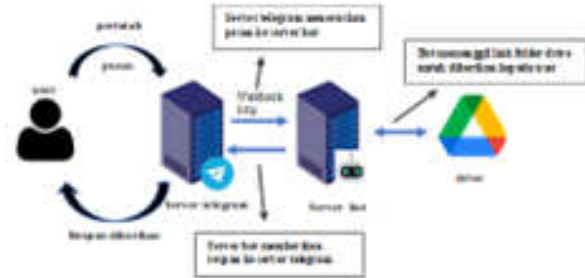
Hardware	Spesifikasi
Laptop	Nama produk : HP 14-AN004AU Notebook Prosesor : AMD Quad-Core A8-7410 APU Grafis : Radeon R5 Graphics (2.2 GHz, up to GHz, 2 MB cache) Memori : 4 GB DDR3L-1600 SDRAM (1 x 4 GB) Kapasitas penyimpanan : 500 GB Display : 14 Inch diagonal HD SVA BrightView LED
Software	Keterangan
Telegram	media informasi aplikasi pemesanan yang menawarkan berbagai keunggulan yaitu fitur bot.
Google drive	layanan penyimpanan berbasis cloud gratis yang memungkinkan pengguna untuk menyimpan dan mengakses file secara online.
Google apps script	platform scripting yang dikembangkan oleh Google untuk pengembangan aplikasi ringan di platform G Suite

2. Design

Tahap ketiga dari penelitian ini ialah Desain. Desain bertujuan untuk membuat sebuah rancangan dalam bentuk skema dan juga *flowchart* untuk dibentuknya sebuah program. Desain memiliki beberapa metode penyusunan. Dan berikut ini ialah beberapa metode dari desain.

a. Metode Komunikasi Webhook

Webhook merupakan salah satu metode komunikasi informasi dalam Bot Engine Telegram. Secara keseluruhan Bot Engine telegram sendiri memiliki 2 metode komunikasi yaitu Long Pooling dan Webhook. Dibandingkan dengan Long pooling. Webhook memiliki mekanisme komunikasi yang lebih singkat sehingga lebih cepat untuk digunakan dalam merespon maupun memproses informasi sebagainya. Berikut ini merupakan gambaran dari metode *webhook*.



Gambar 2. Flow *webhook*

Pengguna pada gambar tersebut digambarkan sebagai client/telegram aplikasi. Ketika pengguna mengirimkan pesan atau perintah maka pesan tersebut akan disimpan di telegram server yang kemudian melewati *webhook* untuk disambungkan dengan bot server yaitu google apps script dengan program yang kita susun didalamnya. Setelah itu data pesan tadi diolah di bot server dan mengambil respon yang sesuai dengan pesan tersebut. Data juga bisa diambil melalui google drive dengan bentuk *link* navigasi yang disematkan di menu mata kuliah. Setelah respon dirasa sudah benar maka respon dikirimkan ke telegram server kembali untuk diteruskan menuju telegram client

b. Alur pembuatan bot

Pembuatan bot memiliki alur yang harus urut dikarenakan setiap proses merupakan atau hal yang melanjutkan metode selanjutnya, jadi apabila proses pertama belum dilakukan maka proses kedua akan terhambat pengejaannya, begitupun seterusnya. Berikut ini merupakan skema dari pembuatan bot.



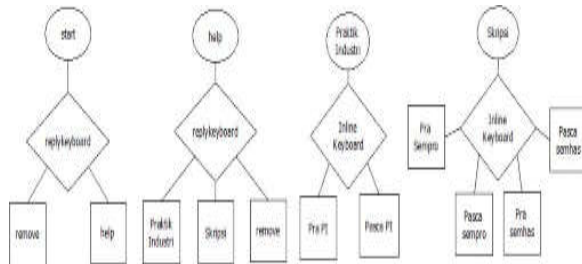
Gambar 3. Alur pembuatan bot

Pembuatan bot dimulai dari pembuatan folder di gdrive yang berisikan file dokumen Praktik industri dan Skripsi, setelah itu *link* folder diubah menjadi .unesa.me lalu copy di

notepad. Selanjutnya memulai penyusunan kode program untuk telegram bot. setelah program telah selesai maka tinggal melakukan deploy dan set *webhook* untuk bot.

c. Desain menu bot

Desain menu yang akan dibuat merupakan desain yang cukup sederhana dan simpel. Ini bertujuan untuk mempermudah pengguna dalam menggunakan bot. untuk perintah /start menggunakan markup *replykeyboard*. Begitu juga perintah /help akan muncul menu keyboard. Berbeda dengan menu mata kuliah praktik industri dan skripsi akan muncul inline keyboard.



Gambar 4. Desain menu

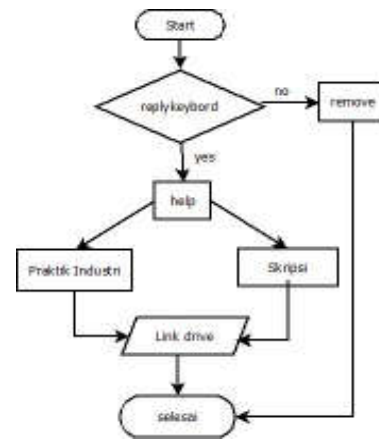
d. Desain Similarity menu

Similarity menu merupakan salah satu fitur tambahan yang ada di dalam bot telegram yang berfungsi sebagai padanan kata dari menu yang ada. Jadi pengguna tidak perlu tidak perlu memanggil perintah yang sama selalu. Dengan menggunakan padanan kata ini pengguna bisa langsung diberikan pesan yang sama.

Tabel 2. Padanan kata sapaan

Kata/perintah	Padanan kata
Sapaan	Hai, halo, hi, hei, assalamualaikum
Praktik industri	Magang, pkl, PI, sempi,
Skripsi	Semhas, sempro, Tugas akhir

e. Cara kerja bot



Gambar 6. Alur kerja bot

Gambar di atas merupakan gambar dari alur kerja bot telegram yang peneliti buat. Alur tersebut dimulai dari *command* start lalu ke help dan ke tujuannya yaitu *link drive* tempat dokumen disimpan. Lalu selesai

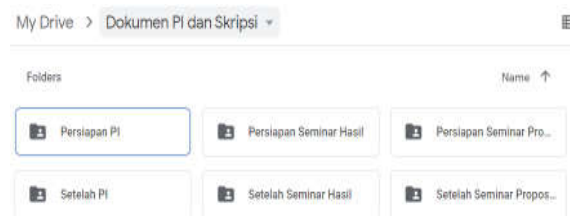
3. Implementation

Tahap selanjutnya ialah implementasi. Di penelitian ini implementasi memiliki 4 tahapan yang diambil dari tahap desain yaitu pembuatan *link* dokumen, pengambilan bot API, penyusunan kode program, serta set *webhook* dan deploy.

a. Pembuatan link dokumen

Pembuatan *link* dokumen dilakukan dengan menggunakan google drive sebagai media penyimpanan. Google drive dianggap sebagai media penyimpanan yang praktis karena tidak menghabiskan memori di perangkat, karena menggunakan sistem cloud. Berikut ini adalah gambaran proses pembuatan *link* dokumen.

b. Pembuatan folder di DRIVE

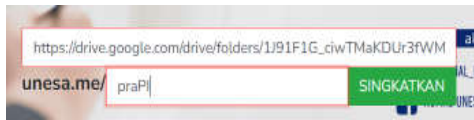


Gambar 7. Folder dokumen

Dokumen – dokumen yang sudah kita kumpulkan dalam analisis kebutuhan kita jadikan dalam satu folder di google drive. Dan gambar diatas merupakan folder – folder yang sudah dibuat oleh peneliti di google drive pribadi.

c. Pergantian link google drive ke link unesa.me

Link yang didapat dari drive disalin lalu ditempel pada kolom *link* unesa.me dan *link* sudah berhasil dirubah.



Gambar 8. Unesame

d. **Daftar link yang dibuat**

Berikut adalah daftar *link* dokumen :

- Unesa.me/prapI
- Unesa.me/pascaPI
- Unesa.me/prasempro
- Unesa.me/prasemhas
- Unesa.me/pascaSemhas

e. **Pengambilan bot API**

Tahap kedua dari implementasi ialah pengambilan bot api. Tahap ini dilakukan dengan tujuan mengambil token dari telegram bot yang dibuat. Token inilah yang nanti digunakan untuk mengatur bot agar melakukan berbagai pekerjaan yang disusun pada kode program. Jika menggunakan metode *webhook*, kode dapat dikirimkan menggunakan *application/json* atau *application/x-www-form-urlencoded* untuk meneruskan parameter. API yang digunakan ialah ialah telegram bot API. Lalu format data yang digunakan adalah JSON karena semua dasar pengiriman data yang digunakan oleh server telegram menggunakan JSON. JSON merupakan format yang menyimpan informasi terstruktur dan biasanya digunakan untuk mentransfer data antara server dan klien.



Gambar 9 .Telegram Bot API

f. **Penyusunan kode program**

Tahap ketiga ialah penyusunan kode program atau *coding*. Disini penditi menggunakan *google apps script* sebagai media utama editing program atau bisa disebut *code editor*. Bahasa yang digunakan untuk menyusun program adalah *javascript cloud scripting*. Peneliti menggunakan library *lumpia* sebagai framework penyusun program. *Lumpia* merupakan sebuah framework buatan salah satu programmer indonesia untuk bot telegram yang digunakan pada *google apps script*. Berikut ini susunan kode program yang dibuat.

g. **Inisiasi fungsi dalam bot**

Kode dibawah ini terdiri dari beberapa inisiasi yaitu *const bot* dan *const token* yang digunakan untuk menginstall fungsi library *lumpia* agar dapat digunakan dalam bot. lalu ada *const button*, *markup*, dan *helper* yang merupakan salah satu fungsi

didalam library *lumpia* agar bisa menggunakan model – model menu dalam bot

```
const token = '5287860770:AAgtD4yWoefHuunR0s3-kF8dJzJVKqFScjK'  
  
const bot = new lumpia.init(token, {log_id:'2101624759'})  
  
const button = lumpia.button,  
markup = lumpia.markup,  
helper = lumpia.helper;
```

Gambar 10. Inisiasibot

h. **Start**

Selanjutnya adalah susunan kode untuk *command /start*, biasanya *start* digunakan untuk memulai sebuah bot jadi ketika pengguna baru masuk kedalam bot maka akan diminta untuk mengklik tombol mulai/*start*.

```
bot.start(ctx => {  
  let nama = ctx.from.first_name;  
  if (ctx.from.last_name) nama += ' ' + ctx.from.last_name;  
  nama = helper.clearHTML(nama);  
  
  let pesan = `Halo <b>${nama}</b>, berkenalkan aku ini bot!<br>silahkan pilih menu dibawah ini`;  
  
  let keyboard= [];  
  keyboard[0] = [  
    button.text('help'),  
    button.text('remove')  
  ];  
  
  ctx.replyWithHTML(pesan, {  
    reply_markup: markup.keyboard(keyboard)  
  });  
})  
  
bot.hears(/remove/i, ctx=>{  
  let pesan = 'keyboard dihapus';  
  ctx.replyWithHTML(pesan, {  
    reply_markup: markup.removeKeyboard()  
  });  
})
```

Gambar 11. Start code

i. **Help**

```
bot.hears(/help/i, ctx=>{  
  let pesan = 'silahkan memilih sesuai kepentingan';  
  let keyboard = [  
    button.text('Praktik Industri'),  
    button.text('Skripsi'),  
    button.text('Remove')  
  ];  
  
  ctx.replyWithHTML(pesan, {  
    reply_markup: markup.keyboard(keyboard, {columns : 3})  
  })  
})
```

Gambar 12. Help code

Kode diatas adalah susunan kode untuk *command /help*, yang berisikan menu mata kuliah yang digunakan dalam penelitian yang ditampilkan dalam bentuk *reply keyboard*

j. **Auto respon menu utama**

Langkah selanjutnya ialah melakukan koding untuk menu utama yaitu menu mata kuliah. Kode yang digunakan hampir

sama dengan perintah /help namun ada perbedaan pada bagian penggunaan keyboard yang dimana menggunakan inline keyboard sebagai model utama.

```
teks = ['praktik industri', 'Praktik industri', 'PI', 'Pi',  
'magang', 'pk1', 'sempi', 'Sempi']  
bot.hears(teks, ctx => {  
  let pesan = 'silahkan memilih pilihan dibawah ini\n\nketerangan _  
  :n1 = pra/sebelum\n2 = pasca/sesudah _  
  let keyboard= [];  
  
  keyboard[0] = [  
    button.url('📄 pi1', 'unesa.me/praPI'),  
    button.url('📄 pi2', 'unesa.me/pascaPI')  
  ];  
  
  ctx.replyWithHTML(pesan, {  
    reply_markup: markup.inlineKeyboard(keyboard)  
  });  
});  
  
kata = ['skripsi', 'Skripsi', 'Tugas akhir', 'Semhas', 'semhas', 'sempro', 'Sempro']  
bot.hears(kata, ctx => {  
  
  let pesan = 'silahkan memilih pilihan dibawah ini\n\nketerangan _  
  :n1 = pra/sebelum\n2 = pasca/sesudah _  
  let keyboard= [];  
  
  keyboard[0] = [  
    button.url('📄 Sempro1', 'unesa.me/praSempro'),  
    button.url('📄 Sempro2', 'unesa.me/pascaSempro')  
  ];  
  keyboard[1] = [  
    button.url('📄 Semhas1', 'unesa.me/praSemhas'),  
    button.url('📄 Semhas2', 'unesa.me/pascaSemhas')  
  ];  
  
  ctx.replyWithHTML(pesan, {  
    reply_markup: markup.inlineKeyboard(keyboard)  
  });  
});
```

Gambar 13. Autorespon menu mata kuliah

k. Similarity word

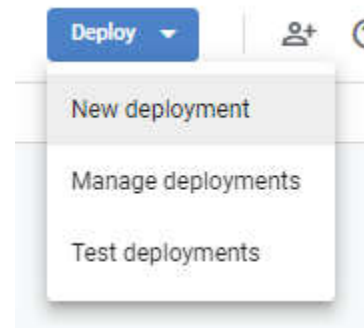
langkah selanjutnya adalah menyusun kode untuk similarity word yang dimana fungsi ini berguna untuk memanggil pesan respon yang sama dengan kata perintah yang menjadi kata utama perintah.

```
bot.hears(['hai', 'hi', 'halo', 'hei'], ctx => ctx.reply('Hai juga!'));  
  
bot.hears(/ass?alamu?ala[yi]+ku+m/i, ctx => ctx.reply('Wa'alaikumussalam Wr Wb'));  
  
teks = ['praktik industri', 'Praktik industri', 'PI', 'Pi', 'magang', 'pk1', 'sempi', 'Sempi']  
bot.hears(teks, ctx => {  
  
  kata = ['skripsi', 'Skripsi', 'Tugas akhir', 'Semhas', 'semhas', 'sempro', 'Sempro']  
  bot.hears(kata, ctx => {
```

Gambar 14. Similarity word code

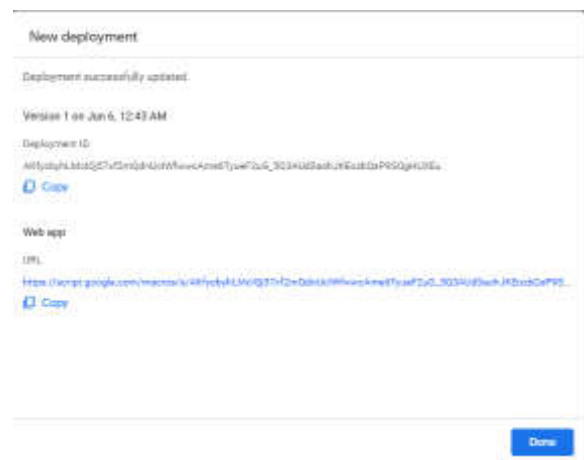
l. Set webhook dan deploy

Tahap terakhir dari implementasi ialah set *webhook* dan deploy. Fungsi dari tahap ini ialah untuk melakukan hosting ke server yang disediakan. Pada kasus kali ini program akan dihosting secara langsung ke server google karena penelitian ini menggunakan google apps script yang sudah terintegrasi langsung dengan server google. Untuk men setting alamat *webhook* pertama – tama yang perlu dilakukan ialah melakukan deploy ke server untuk mendapatkan alamat http nya.



Gambar 15. Menu deployment

Lakukan deployment untuk kode program yang sudah tersusun, dan ikuti semua prosedur mulai dari penzinan ke semua lalu perizinan google apps script untuk mengakses drive sampai akhirnya mendapat alamat http sebagai alamat *webhook*



Gambar 16. A alamat http

Setelah alamat didapatkan, masukkan alamat tersebut kedalam kode untuk dilakuakn fungsi *setWebhook*. *Setwebhook* bisa dilakukan melalui kode pemrograman maupun dilakukan dengan secara manual dengan mengetikkan alamat *api.telegram.org/bot(token)/setWebhook=url*.



Gambar 17. setWebhook

4. Verification

Tahap penelitian yang selanjutnya ialah verification/pengujian. Tahap terakhir dari penelitian ini adalah pengujian. Pengujian akan dilakukan dengan 2 tahap yaitu pengujian infrastruktur dan juga pengujian aplikasi. Berikut

ada;ah penjelasan mengenai kedua tahap yang akan digunakan sebagai metode pengujian.

Pengujian infrastruktur digunakan untuk mengecek kedalaman struktur bot yang dibuat mulai dari cek dan uji *webhook*, uji kecepatan respon bot saat menggunakan metode *webhook*, pengujian kecepatan respon yang dikirim saat pengujian multi client chat. Lalu ada pengujian aplikasi yang menguji fungsionalitas menu standar dan pengujian *similarity word* pada bot telegram.

5. Maintenance

Tahap terakhir dari penelitian ini adalah maintenance, yang artinya pemeliharaan. Pemeliharaan perlu dilakukan untuk menguji bot apakah masih aktif atau tidak. Lalu apakah ada saran tambahan menu, fungsi, dll dari pengguna. Pemeliharaan yang baik akan membuat bot semakin baik kedepannya dengan melakukan pemeliharaan kedalam beberapa struktur bot seperti bagian kode program apakah ada kesalahan, lalu apakah *webhook* sudah diset dan apakah bot sudah bisa merespon pengguna ketika melakukan operasi.

III. HASIL DAN PEMBAHASAN

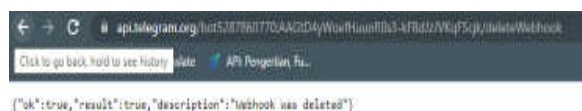
Pengujian dilakukan menggunakan perangkat laptop dan juga smartphone milik peneliti dan beberapa partisipan yang diminta bantuan dalam pengujian. adapula penggunaan aplikasi POSTMAN untuk pengujian metode komunikasi dari bot. Seperti yang dijelaskan di bagian metode penelitian, pengujian akan dilakukan dengan 2 macam jenis yaitu infrastruktur dan aplikasi.

1. Pengujian Infrastruktur

Pengujian ini menggunakan aplikasi POSTMAN sebagai media utama untuk mencari informasi terkait pengujian yang dilakukan. Peneliti akan melakukan metode POST untuk melakukan pengiriman data ke server lalu diteruskan ke bot. respon yang diterima tadi lah yang akan dijadikan data pada pengujian ini.

m. Cek dan uji *webhook*

Pengujian ini dilakukan untuk mengecek apakah *webhook* sudah diset dan sudah bisa menjadi penghubung antar server. Pengujian akan dilakukan secara manual menggunakan website api telegram dan menggunakan aplikasi POSTMAN.



Gambar 18. Delete*webhook*

Gambar diatas menunjukkan fungsi *deleteWebhook* yang dilakukan secara manual menggunakan website

[https://api.telegram.org/bot\(token\)/deleteWebhook](https://api.telegram.org/bot(token)/deleteWebhook). Ini digunakan untuk menghapus *webhook* yang tersambung antara server telegram dan bot server yang kemudian bisa dipasang *http webhook* yang baru. Jika *webhook* belum terpasang maka bot tidak akan berfungsi.



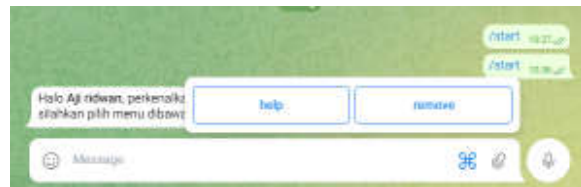
Gambar 19. Perintah tidak berfungsi

Seperti yang terjadi pada gambar diatas perintah yang dikirimkan oleh pengguna tidak diberikan respon oleh bot dikarenakan telegram belum tersambung dengan bot server google script.



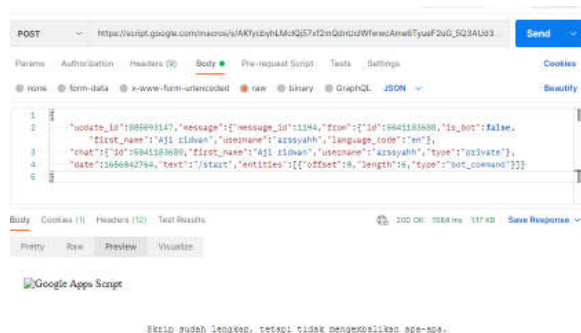
Gambar 20. set*Webhook*

Setelah mendapatkan alamat *http webhook* dari kode yang dideploy tadi. Alamat tersebut disali lalu ditempel ke alamat [https://api.telegram.org/setWebhook\(token\)/setWebhook?url=\(url\)](https://api.telegram.org/setWebhook(token)/setWebhook?url=(url)) maka hasil yang didapat adalah *webhook* sudah terset atau sudah tersambung.



Gambar 21. Respon bot

Jika *webhook* sudah disambungkan maka bot akan memberikan respon kepada pengguna yang artinya telegram server dan bot server sudah tersambung menggunakan metode *webhook*. Hal ini juga bisa diujikan melalui aplikasi POSTMAN dengan cara mengirimkan pesan json kepada alamat *http* menggunakan request POST.



Gambar 22. POST /start command

Gambar diatas menunjukkan pengiriman request berupa POST menuju pengguna dengan perintah /start. Jika *webhook* sudah ter set maka respon akan langsung diberikan oleh bot jika belum maka bot juga tidak akan memberikan respon apapun.



Gambar 23. Responbot setelah uji POSTMAN

Gambar diatas menunjukkan respon yang diberikan oleh bot untuk perintah /start. Disini terlihat tidak ada perintah /start yang dikirim melalui aplikasi telegram namun perintah dikirim melalui aplikasi POSTMAN. Ini menunjukkan jika alamat yang diberikan oleh google script merupakan *webhook* http yang benar.

n. Kecepatan auto respon bot

Berdasarkan pengujian metode POST yang dilakukan menggunakan aplikasi POSTMAN kecepatan respon dari bot berubah – rubah tidak dalam waktu yang sama namun jarak yang diberikan tidak terlalu jauh. Berikut tabel kecepatan respon dari tiap menu yang dijalankan.

Tabel3. Time responses

Menu	Start	Help	Praktik industri	Skripsi
Waktu respon	1610 ms	1014 ms	1250 ms	1263 ms

Tabel diatas merupakan tabel kecepatan respon yang diberikan oleh bot kepada pengguna saat membalas perintah yang diberikan. Saat pengguna mengirimkan perintah start maka respon akan diberikan dalam waktu 1610ms yang dimana sama dengan 1,61 s yang bisa dibilang cukup cepat. Lalu ketika memberikan perintah /help maka respon diberikan dalam waktu 1014ms, begitupun saat memberikan perintah menu praktik industri dan skripsi bot mampu membalas dengan kecepatan respon 1250ms dan 1263ms. Kecepatan respon yang diberikan tidak selalu sama untuk setiap perintahnya, namun masih dalam skala dibawah 2s yang mana cukup cepat untuk sebuah respon bot.

o. Tes multi client chat

Pengujian ini dilakukan menggunakan perangkat mobile milik pengguna. Penggunaan mobile lebih dikedepankan dikarenakan lebih simple dan pasti semuanya membuka telegram melalui perangkat mobile masing – masing. Berikut

adalah tabel respon perangkat dituliskan dalam bentuk ms (*milisecond*).

Tabel 4. Tabel respon multi chat 1

Per	Start	Help	PI	Skripsi
1.	1610 ms	1014 ms	1250 ms	1263 ms
2.	1186 ms	1405 ms	1312 ms	1247 ms
3.	1499 ms	1154 ms	1237 ms	1099 ms
4.	1589 ms	1205 ms	1153 ms	1131 ms
5.	1174 ms	1211 ms	1258 ms	1190 ms

Nb : per= perangkat

Tabel diatas merupakan hasil dari dijalankannya perintah dari banyak klien terhadap bot. waktu yang tertera didalamnya merupakan isi dari waktu respon yang terjadi saat klien memberikan perintah kepada bot. pengujian diatas dilakukan menggunakan aplikasi POSTMAN dengan mengambil data yang dikirimkan oleh pengguna kepada bot melalui mode *debugging* pada *google script*. Pengujian ini dilakukan untuk mengetahui apakah bot mampu dijalankan oleh banyak pengguna sekaligus atau tidak. Dengan hasil yang tertera dalam tabel menunjukkan jika bot tidak mengalami delay atau apapun selama lebih dari 2s. untuk keterangan tabel diatas ialah ketika tiap pengguna memberikan perintah start, respon yang diberikan oleh bot ada di angka 1610ms, 1186ms, 1499ms, 1589ms, dan 1174ms. Yang dimana semuanya ada dalam kisaran 1second lebih.

2. Pengujian Aplikasi

Pengujian aplikasi digunakan untuk mengetes kesesuaian menu dengan kode program yang dibuat, lalu untuk memeriksa apakah ada perintah yang tidak mampu dijalankan. Pengujian dilakukan menggunakan perangkat pribadi peneliti.

a. Pengujian fungsionalitas menu

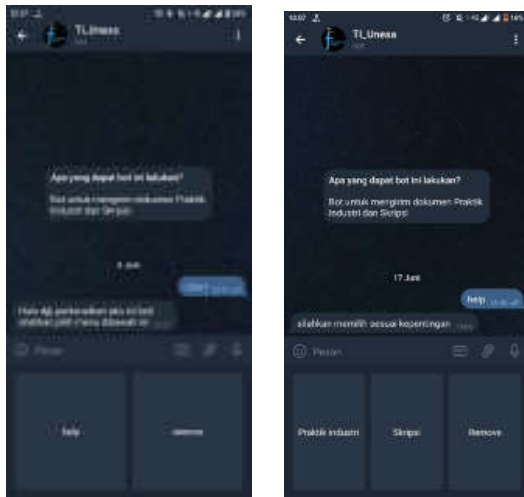
Pengujian ini dilakukan untuk mengetes tiap – tiap perintah yang ada di dalam bot. dalam pengujian ini terdapat penerapan model *classification* dan *similarity word*. *Classification* terdiri dari menu utama, menu mata kuliah. Lalu *similarity word* terdiri dari menu praktik industri dan menu skripsi

1). Classification

a). Menu utama

Menu utama terdiri dari 2 perintah utama yaitu /start dan juga. Perintah start digunakan untuk memulai sebuah bot atau bisa difungsikan sebagai permulaan bot. Setelah tombol start/mulai diklik maka akan muncul menu keyboard yang berisikan perintah HELP dan REMOVE. Perintah help digunakan untuk menampilkan mata kuliah yang disediakan oleh bot telegram. menggunakan fungsi yang sama yaitu menu

keyboard seperti perintah START. Isi dari keyboard tersebut ialah menu Praktik Industri dan juga Skripsi.



Gambar 24. Perintah/Start dan help

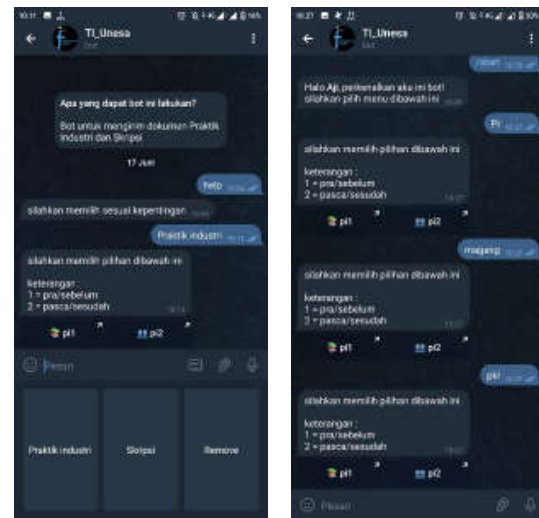
b). Menu mata kuliah

Menu ini akan dijelaskan sekaligus pada model *similarity word* karena mendapat 2 model sekaligus.

2). *Similarity word*

a). Menu mata kuliah praktik industri

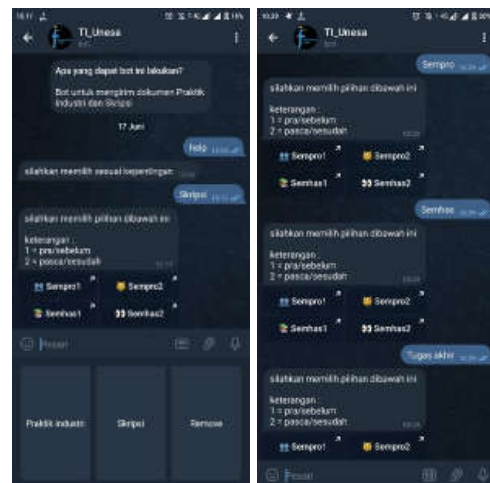
Menu praktik industri berfungsi untuk menampilkan *link* folder tempat dokumen praktik industri berada. Ketika pengguna memilih menu praktik industri maka akan muncul tampilan inline keyboard dengan keterangan yang terdapat di pesan respon bot. ketika pengguna meng klik pilihan pada keyboard maka akan muncul pesan pop – up untuk menavigasikan pengguna menuju folder tempat dokumen berada. Lalu menu praktik industri juga memiliki padanan kata yang mana ketika pengguna mengetikkan kata tersebut maka akan muncul pesn yang sama seperti menu utama praktik industri.



Gambar 25. Menu praktik industri dan padanan katanya

b). Menu mata kuliah Skripsi

Menu skripsi berfungsi untuk menampilkan *link* folder tempat dokumen praktik industri berada. Ketika pengguna memilih menu skripsi maka akan muncul tampilan inline keyboard dengan keterangan yang terdapat di pesan respon bot. ketika pengguna meng klik pilihan pada keyboard maka akan muncul pesan pop – up untuk menavigasikan pengguna menuju folder tempat dokumen berada. Lalu menu skripsi juga memiliki padanan kata yang mana ketika pengguna mengetikkan kata tersebut maka akan muncul pesan yang sama seperti menu utama skripsi.



Gambar 26. Menu skripsi dan padanan katanya

IV. KESIMPULAN

Berdasarkan hasil pengujian penelitian sebelumnya, maka dapat ditarik beberapa kesimpulan, yaitu :

1. Sistem *auto message* sudah berhasil dibuat dan diterapkan dengan aplikasi telegram dengan pengujian ketika pengguna memberikan perintah atau beberapa kata maka bot akan langsung membalas pesan tersebut dengan data yang didapat pada bot server.
2. Metode *webhook* berhasil diimplementasikan terhadap bot, sehingga bot mampu berjalan secara otomatis karena telegram server sudah tersambung dengan bot server melalui *webhook*.
3. Dokumen yang disediakan sudah dijadikan dalam satu folder drive yang bisa diakses oleh pengguna ketika menggunakan bot. Bot akan mengirimkan pesan pop – up berupa *link* unesa.me (hasil dari perubahan *link* gdrive) agar pengguna dapat memasuki folder tersebut.
4. Untuk pengujian *classification* dan *similarity word* diperoleh hasil sebagai berikut :
 - Penggunaan *classification* berhasil digunakan untuk membedakan setiap menu pada bot.
 - Penggunaan *similarity word* juga mampu diadaptasikan ke dalam bot dengan memberikan padanan kata kepada menu matakuliah.

V. SARAN

Dari hasil penelitian yang sudah didapatkan, peneliti memiliki beberapa saran untuk membuat bot telegram bisa dijadikan sebagai bot official milik jurusan dengan melakukan beberapa pengujian kembali dan meminta perizinan agar telegram bot bisa digunakan oleh mahasiswa jurusan teknik informatika Unesa. Serta bot bisa dikembangkan lagi menjadi lebih baik dengan menambahkan beberapa fitur yang lebih lengkap seperti dapat mencari info dosen, staff dll. Lalu dapat menjadi sarana pengumpulan dokumen untuk pihak tata usaha agar lebih efisien dalam hal pengurusan administrasi.

VI. UCAPAN TERIMA KASIH

Ujaran rasa syukur dan terima kasih penulis sampaikan kepada :

1. Allah SWT atas segala rahmat dan hidayah yang diberikan sehingga penelitian bisa diselesaikan
2. Bapak Agus Prihanto S.T., M.Kom selaku dosen pembimbing yang senantiasa memberikan bantuan dari awal hingga akhir sehingga saya bisa menyelesaikan penelitian ini.
3. Bapak Purwadi selaku ayah yang senantiasa memberikan doa yang terbaik untuk saya
4. Almarhum ibu Supami selaku ibu saya yang mendoakan saya di akhirat sana.
5. Candra purwaningrum selaku saudara saya yang memberikan bantuan secara mental
6. Ardian, wildan dan yuniar selaku teman saya yang mau memberikan bantuan pada penelitian ini

VII. REFERENSI

- [1]. Cholik, Cecep. "Pemanfaatan Teknologi Informasi Dan Komunikasi Untuk Meningkatkan Pendidikan Di Indonesia." *Syntax Literate; Jurnal Ilmiah Indonesia* [Online], 2.6 (2017): 21 - 30. Web. 24 May. 2022
- [2]. M. Aris Widya and 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.
- [3]. Saribekyan, Hayk and Margvelashvili, Akaki. 2017."Security Analysis of Telegram". Diakses tanggal 24 Mei 2022
- [4]. Heri Khariono, Rizky Parlita, Haidar Ananta Kusuma, and Dimas Arif Setyawan, "Pemanfaatan Bot Telegram Sebagai E-Learning Ujian Berbasis File", *JIP*, vol. 7, no. 4, pp. 65-72, Aug. 2021.
- [5]. Nufusula, R., and Susanto, A. (2018). *Rancang Bangun Chat Bot Pada Server Pula Menggunakan Telegram Bot Api*. *Jurnal of Information System*, 80–88. <https://doi.org/10.33633/joins.v3i1.1884>
- [6]. J. K. Adangbain and E. S. Bata, "Pemanfaatan Bot Telegram Sebagai Media Informasi Dan Layanan Akademik Dengan Metode Webhook", *semmau*, pp. 106-112, Dec. 2021.
- [7]. A. Prayogi. and Nurjayadi, "Bot Permainan Tebak Gambar Pengenalan Adat Istiadat Riau Pada Apl Ikasi Line Dengan Metode Webhooks," vol. 6, no. 2, pp. 161167, Oktober 2018