

**PENERAPAN *STRING MATCHING* DAN *REGULAR EXPRESSION*
DALAM PEMBANGUNAN *DEADLINE REMINDER ASSISTANT***

LAPORAN TUGAS BESAR 3

Diajukan sebagai salah satu Tugas Besar 3

IF2211 Strategi Algoritma Semester II tahun 2020/2021

Oleh:

Faris Hasim Syauqi (13519050)

Mgs. Tabrani (13519122)

Muhammad Rizal Muhaimin (13519136)



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2021**

BAB I

DESKRIPSI TUGAS

Dalam tugas besar ini, Anda akan diminta untuk membangun sebuah chatbot sederhana yang berfungsi untuk membantu mengingat berbagai deadline, tanggal penting, dan task-task tertentu kepada user yang menggunakannya. Dengan memanfaatkan algoritma String Matching dan Regular Expression, Anda dapat membangun sebuah chatbot interaktif sederhana layaknya Google Assistant yang akan menjawab segala pertanyaan Anda terkait informasi deadline tugas-tugas yang ada.

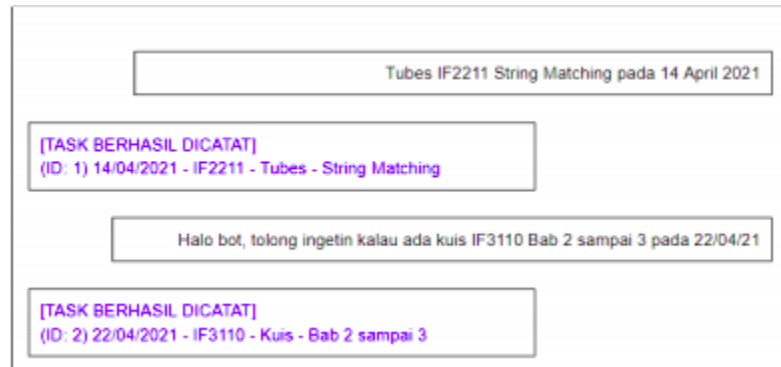
Fitur-Fitur Aplikasi:

Deadline Reminder Assistant. akan dibangun dengan sistem Question and Answer dimana pengembang diharapkan sudah menyediakan kumpulan formula tertentu untuk melakukan pendeteksian setiap perbedaan command atau perintah pada aplikasi Chatbot. Berikut ini adalah runtutan fitur yang dimiliki oleh Deadline Reminder Assistant tersebut.

1. Menambahkan task baru
 - a. Suatu kalimat diklasifikasikan sebagai suatu task apabila mengandung semua komponen berikut ini:
 - i. Tanggal (format dibebaskan)
 - ii. Kode Mata Kuliah / Nama Mata Kuliah (dibebaskan)
 - iii. Jenis Tugas (berdasarkan daftar kata penting yang sudah disediakan)
 - iv. Topik Tugas (tidak ada batasan)
 - b. Point i sampai dengan iv diklasifikasikan menggunakan regular expression sehingga masukan kalimat benar-benar layaknya kalimat sehari-hari
 - c. Jika pesan berhasil dikenali oleh assistant, maka assistant akan mengirim pesan balasan yang berisi ID (sesuai urutan task diinput), tanggal, kode mata kuliah, jenis tugas, dan topik tugas. Contoh pesan balasan dari bot sebagai berikut.

[TASK BERHASIL DICATAT]
(ID: 1) 14/04/2021 - IF2211 - Tubes - String matching

d. Contoh interaksi



2. Melihat daftar task yang harus dikerjakan

- a. Seluruh task yang sudah tercatat oleh assistant
 - Contoh perintah yang dapat digunakan: “Apa saja deadline yang dimiliki sejauh ini?”
- b. Berdasarkan periode waktu
 - i. Pada periode tertentu (DATE_1 until DATE_2) Contoh perintah yang dapat digunakan: “Apa saja deadline antara DATE_1 sampai DATE_2?”
 - ii. N minggu ke depan Contoh perintah yang dapat digunakan: “Deadline N minggu ke depan apa saja?”
 - iii. N hari ke depan Contoh perintah yang dapat digunakan: “Deadline N hari ke depan apa saja?”
 - iv. Hari ini Contoh perintah yang dapat digunakan: “Apa saja deadline hari ini?”
- c. Berdasarkan jenis task (kata penting)
 - i. Sesuai dengan daftar task yang didefinisikan
 - ii. User dapat melihat daftar task dengan jenis task tertentu
 - iii. Misalnya: “3 minggu ke depan ada kuis apa saja?”, maka Chatbot akan menampilkan daftar kuis selama 3 minggu kedepan

Catatan: Eksekusi perintah pengguna bisa mencakup ketiga poin sekaligus sehingga formula pengenalan command sebaiknya dibuat sebagai satu kesatuan utuh.

Contoh interaksi

The screenshot shows a chat window with the following interactions:

- User:** Apa saja deadline yang dimiliki sejauh ini?
- System:**
[Daftar Deadline]
1. (ID: 1) 14/04/2021 - IF2211 - Tubes - String Matching
2. (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3
- User:** Apa saja deadline antara 03/04/2021 sampai 15/04/2021?
- System:**
[Daftar Deadline]
1. (ID: 2) 14/04/2021 - IF3110 - Kuis - Bab 2 sampai 3
- User:** Deadline 3 minggu ke depan apa saja?
- System:**
[Daftar Deadline]
1. (ID: 1) 14/04/2021 - IF2211 - Tubes - String Matching
2. (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3
- User:** Deadline 1 hari ke depan apa saja?
- System:**
Tidak ada
- User:** 3 minggu ke depan ada kuis apa saja?
- System:**
[Daftar Deadline]
1. (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

Keterangan penting:

- Perintah yang digunakan pengguna bisa tidak selalu sama, asalkan mengandung kata kunci yang ditentukan (kata kunci tiap perintah bisa ditentukan sendiri). Misal kedua contoh di bawah ini memberikan output yang sama
 - Apa saja deadline antara 03/04/2021 sampai 15/04/2021?
 - Antara 03/04/2021 dan 15/04/2021 ada deadline apa saja ya?

3. Menampilkan deadline dari suatu task tertentu

- a. Hanya berlaku untuk task yang bersifat Tugas atau memiliki tenggat waktu
- b. Misalnya: “Deadline tugas IF2211 itu kapan?”
- c. Contoh interaksi

The image shows a chatbot interface with a white background and a thin black border. Inside, there are two rectangular boxes. The top box is white with a thin black border and contains the text "Deadline tugas IF2211 itu kapan?". The bottom box is also white with a thin black border and contains the date "14/04/2021" in a purple font.

4. Memperbaharui task tertentu

- a. Memperbarui tanggal dari suatu task (dalam kehidupan nyata, tentu ada kejadian dimana deadline dari suatu task diundur)
- b. Perintah yang dimasukkan meliputi 1 keyword untuk memperbaharui suatu task dan nomor task tertentu.
- c. Misalnya:
 - “Deadline task X diundur menjadi 28/04/2021” dimana X merupakan nomor ID dari suatu task.
- d. Apabila task berhasil diperbaharui, Chatbot akan menampilkan pesan sukses memperbaharui suatu task. Sebaliknya, Chatbot akan menampilkan pesan error apabila task yang dimaksud tidak dikenali oleh Chatbot (belum masuk ke dalam Daftar Task)

5. Menandai bahwa suatu task sudah selesai dikerjakan

- a. Apabila user sudah menyelesaikan suatu task, maka task tersebut bisa ditandai bahwa task tersebut sudah selesai dan tidak perlu lagi ditampilkan pada Daftar Task selanjutnya.
- b. Misalnya:
 - “Saya sudah selesai mengerjakan task X” dimana X merupakan nomor ID dari suatu task.
- c. Apabila perintah yang dimasukkan user bisa dieksekusi, Chatbot akan menampilkan pesan sukses. Sebaliknya, Chatbot akan menampilkan pesan error apabila task yang dimaksud tidak dikenali oleh Chatbot (belum masuk ke dalam Daftar Task)

6. Menampilkan opsi help yang difasilitasi oleh assistant

- a. Berisikan command-command yang dapat digunakan oleh user
- b. Misalnya: “Apa yang bisa assistant lakukan?”

- c. Bot akan memberikan hasil berupa daftar kata-kata yang bisa digunakan untuk menambahkan dan melihat daftar task (setiap kelompok bebas membentuknya seperti apa)
- d. Contoh interaksi

Apa yang bisa assistant lakukan?

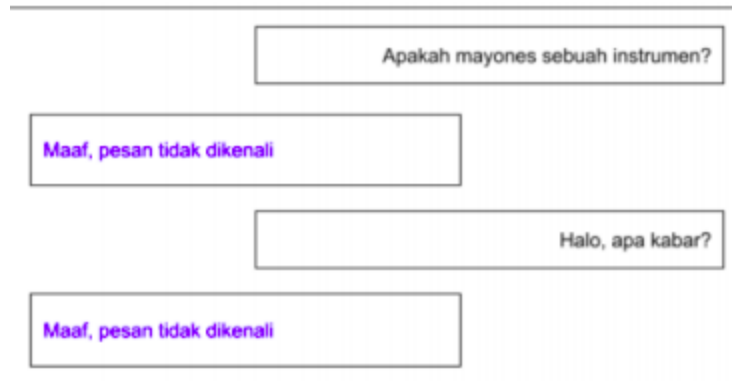
[Fitur]

1. Menambahkan task baru
2. Melihat daftar task
3.
4. (dan seterusnya)

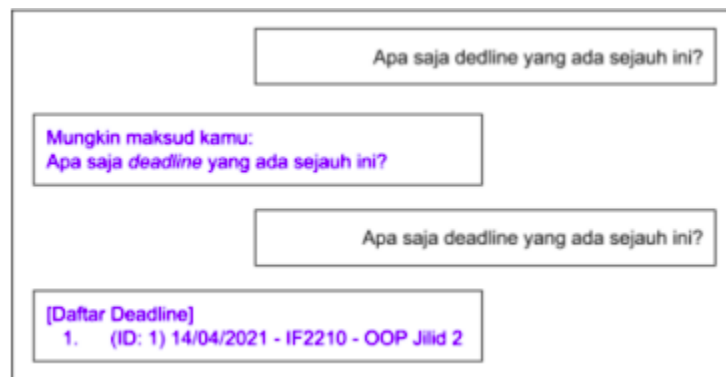
[Daftar kata penting]

1. Kuis
2. Ujian
3. Tugil
4. Tubes
5.

- 7. Mendefinisikan list kata penting terkait apakah itu merupakan suatu task atau tidak
 - a. Minimal terdapat 5 kata penting berbeda, contohnya adalah: ["Kuis", "Ujian", "Tugil", "Tubes", "Praktikum"]
 - b. Kata penting akan digunakan pada penentuan jenis tugas dari suatu task.
 - c. Daftar kata penting tidak perlu dibuat dinamis, cukup static saja atau hardcoded.
- 8. Menampilkan pesan error jika assistant tidak dapat mengenali masukan user.
 - a. Masukan yang tidak termasuk ke dalam jenis pesan di poin 1 sampai 4 dapat dikategorikan sebagai masukan tak dikenali.
 - b. Error message dibebaskan sesuai kreativitas mahasiswa
 - c. Contoh interaksi



9. **(Bonus)** Chatbot dapat memberikan rekomendasi kata jika terdapat kesalahan kata (typo) pada perintah yang ditulis pengguna
- Berikan rekomendasi kata jika perintah masukan pengguna mismatch dengan daftar kata yang diterima chatbot, namun masih memiliki tingkat kemiripan di atas 75%.
 - Contoh interaksi



- Ada berbagai matriks yang dapat dimanfaatkan untuk mencari kemiripan kata, salah satunya adalah Levenshtein distance yang diukur melalui pendekatan dynamic programming

BAB II

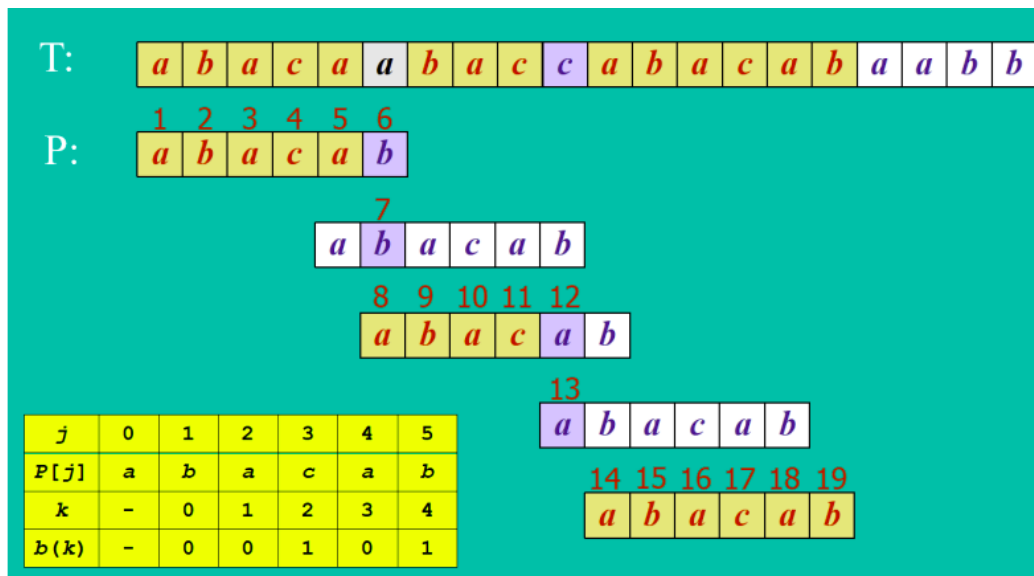
LANDASAN TEORI

A. Deskripsi Singkat Algoritma KMP, BM, dan Regex

Pattern Matching memiliki banyak implementasi dalam kehidupan sehari-hari, seperti pencarian suatu kata dalam sebuah *file* dan mesin pencarian Google. Konsep *pattern matching* adalah mencocokkan sebuah *pattern* dengan sebuah teks. *Pattern* merupakan *substring* dari sebuah teks yang ingin dicari keberadaan *pattern*-nya. Algoritma Knuth-Morris-Pratt (KMP), Boyer-Moore, dan *Regular Expression* merupakan algoritma-algoritma yang dapat dimanfaatkan dalam *pattern matching*.

1. Knuth-Morris-Pratt (KMP)

Algoritma Knuth-Morris-Pratt (KMP) akan mencari *pattern* di dalam teks dengan urutan dari kiri ke kanan dengan melakukan *shifting*. Berikut ini contoh penerapan KMP.

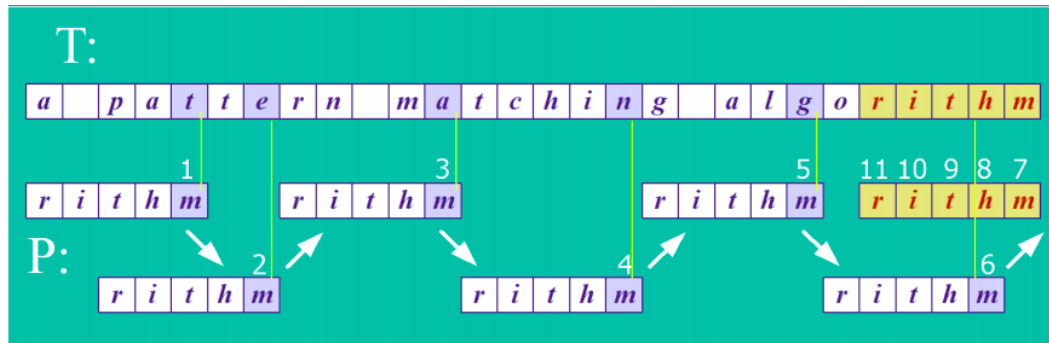


Jumlah perbandingan karakter: 19 kali

Keuntungan dalam menggunakan algoritma KMP adalah algoritma ini tidak melakukan gerakan mundur, sehingga cocok dalam memproses *file* atau teks yang memiliki ukuran besar. Kekurangan dari algoritma ini adalah kurang baik dalam kasus yang memiliki variasi alfabet yang banyak.

2. Boyer-Moore

Algoritma Boyer-Moore menerapkan dua teknik dalam melakukan *pattern matching*, yaitu *looking-glass technique* dan *character-jump technique*. Berikut ini contoh penerapan algoritma Boyer-Moore.



Jumlah perbandingan karakter: 11 kali

Algoritma Boyer-Moore akan memproses dengan cepat apabila variasi alfabetnya besar dan akan memproses dengan lambat apabila variasi alfabetnya kecil.

3. Regular Expression

Regular Expression adalah notasi standar yang mendeskripsikan suatu pola (*pattern*) berupa urutan karakter atau *string*. *Regular Expression* digunakan untuk pencocokan *string* dengan efisien. *Regular Expression* sudah menjadi standar yang tersebar di semua *tools* dan bahasa pemrograman sehingga penting untuk dipelajari. Berikut ini contoh penerapan *Regular Expression* dalam pencocokan *string*.

```
lulusan .*[ii]nformatika [Ii][Tt][Bb]
```

Sebagai seorang lulusan Teknik Informatika ITB, saya malu sistem registrasi di ITB seperti ini. Apa bedanya dengan tahun 1990an?

Pernah nemu ada yg ngaku lulusan Informatika ITB. Gw tanya sana-sini gk ada yg kenal. Eh tauny dia ngelampirin sertifikat training HMIF ITB.

Siapa kak ? Siapa ? "@niaputrinia: Umurnya 26 tahun, lulusan teknik informatika itb. Sekarang kerja jd fotografer dan kontributornya National"

B. Penjelasan Singkat Mengenai *Chatbot*

Istilah *chatbot* dengan nama lain *chatterbot* merupakan sebuah layanan obrolan robot atau tokoh virtual dengan kecerdasan buatan (*Artificial Intelligence*) yang menirukan percakapan manusia melalui pesan suara, obrolan teks ataupun keduanya. Singkatnya *chatbot* merupakan percakapan virtual, yaitu satu pihak adalah sebuah robot *chat*. Fitur *chatbot* ini telah digunakan di berbagai industri untuk menyampaikan informasi.

Chatbot menggunakan berbagai teknologi yang mutakhir di dalamnya, seperti *Artificial Intelligent* (AI), *Machine Learning* (ML), *Deep Learning*, dan *Natural Language Processing* (NLP). *Machine Learning* dapat diaplikasikan sebagai mesin yang mempelajari, menganalisa dan mengenal bermacam bahasa pada *chatbot*. Sementara *Natural Language Processing* (NLP) memiliki kemampuan untuk mengerti dan memahami bahasa manusia lalu memberikan respon yang sesuai dengan bahasa yang digunakan pengguna *chatbot*.

BAB III

ANALISIS PENYELESAIAN MASALAH

A. Langkah Penyelesaian Masalah Setiap Fitur

1. Menambahkan *Task* Baru

Fitur ini memanfaatkan *string matching* dalam menyaring sebuah *pattern* dengan kalimat yang dalam hal ini pesan pengguna sehingga pesan dapat dapat dikenali sebagai suatu task baru atau tidak. Dalam penerapannya perlunya pendefinisian kata kata penting yang menjadi dasar pengambilan keputusan apakah task bisa ditambahkan atau tidak. Kata penting yang didefinisikan dalam tugas kali ini adalah “tubes”, “tucil”, “ujian”, “kuis” dan “pr” serta kata “pada” dan “tanggal” sebagai kunci pemisah antara kata penting dengan tanggal yang ingin dicatat. Pengambilan tanggal dilakukan dengan regular expression sedemikian rupa sehingga format tanggal sesuai dengan yang didefinisikan pada basis data. Format yang dikenali dalam penambahan task baru dalam tubes ini, harus ada kata penting (jenis) yang diikuti nama atau kode mata kuliah kemudian topik dari mata kuliah diikuti dengan kata hubung “pada” atau “tanggal” kemudian tanggal. Sebagai contoh “bot, ingatkan tubes IF2211 String matching pada 28 april 2021.

2. Melihat Daftar *Task* yang Harus Dikerjakan

a. Seluruh task yang sudah tercatat oleh assistant

Fitur ini memanfaatkan *string matching* untuk mengecek apakah ada kata penting “deadline” dan “sejauh” . ketika kata kunci ada dalam kalimat pengguna maka program akan meminta basis data yang telah didefinisikan untuk memanggil data yang sesuai.

b. Berdasarkan periode waktu

1. Pada periode waktu tertentu.

Dengan *string matching*, program akan mencari kata “antara” sebagai kunci dari fitur menampilkan data pada rentang waktu tertentu. Ketika

kata “antara” terdapat dalam kalimat maka regular expression akan menentukan posisi tanggal dan menyesuaikan format tanggal yang sesuai pada fungsi yang ada pada basis data untuk memanggil data yang sesuai.

Dengan memanfaatkan

2. N minggu ke depan

Fungsi *string matching* akan mencari kata penting “minggu” dan “depan” dalam inputan pengguna kemudian mendapatkan kata kunci N minggu sebagai data yang akan dimasukkan dalam fungsi yang menghasilkan tanggal untuk N minggu kedepan. Setelah didapatkan maka dapat dilakukan pemanggilan data yang ada di basis data untuk menampilkan output yang diharapkan.

3. N hari kedepan

Fungsi *string matching* akan mencari kata penting “hari” dan “depan” dalam inputan pengguna kemudian mendapatkan kata kunci N minggu sebagai data yang akan dimasukkan dalam fungsi yang menghasilkan tanggal untuk N minggu kedepan. Setelah didapatkan maka akan dilakukan pemanggilan data yang ada di basis data untuk menampilkan output yang diharapkan.

4. Hari ini

Fungsi *string matching* akan mencari kata penting “hari” dalam inputan pengguna kemudian mendapatkan kata kunci. Setelah didapatkan maka dapat dilakukan pemanggilan data yang ada di basis data untuk menampilkan output yang diharapkan.

c. Berdasarkan jenis kata penting

Fitur ini merupakan fitur modifikasi dari penerapan *string matching* pada poin A.2.b berdasarkan periode waktu tertentu yang menambahkan pengecekan dengan *string matching* apakah inputan pengguna ada satu kata dari kata penting “tubes”,

“tucil”, “ujian”, “kuis” dan “pr”. Setelah didapatkan ,maka akan dilakukan pemanggilan data yang ada di basis data untuk menampilkan output yang diharapkan.

3. Menampilkan *Deadline* dari Suatu *Task* Tertentu

Fitur ini memanfaatkan *string matching* dengan menyaring sebuah *pattern* dengan pesan dari pengguna. *Pattern* yang digunakan adalah *when*, *deadline*, dan kapan. Kemudian, program akan melakukan *pattern matching* jenis tugas yang ingin dicari. Jika *pattern*-nya adalah tugas, program akan langsung melakukan *string matching* dengan mata kuliah terkait. Jika *pattern*-nya bukan tugas, program akan melakukan *string matching* jenis tugas, lalu melakukan *string matching* mata kuliah.

4. Memperbaharui *Task* Tertentu

Fitur ini memanfaatkan *string matching* dengan menyaring sebuah *pattern* dengan pesan dari pengguna. *Pattern* yang digunakan adalah *diundur..* Kemudian program akan mencocokkan pesan dari pengguna yang mengandung ID lalu, mencari tanggal yang ada dimasukan pengguna sebagai data pembaruan dan memperbarui data dengan fungsi yang sudah didefinisikan pada basis data.

5. Menandai Bahwa Suatu *Task* Sudah Selesai Dikerjakan

Fitur ini memanfaatkan *string matching* dengan menyaring sebuah *pattern* dengan pesan dari pengguna. *Pattern* yang digunakan adalah *done*, selesai, dan sudah. Kemudian program akan mencocokkan pesan dari pengguna yang mengandung ID dari *task* yang sudah dikerjakan. Lalu, status dari *task* tersebut akan diubah menjadi sudah diselesaikan di basis data yang dibuat.

6. Menampilkan Opsi *Help* yang Difasilitasi oleh *Assistant*

Fitur ini memanfaatkan *string matching* dengan menyaring sebuah *pattern* dengan pesan dari pengguna. *Pattern* yang digunakan adalah bisa, lakukan, *help*, *command*, dan fitur. Kemudian program akan menampilkan daftar perintah atau fitur yang disediakan program dan menampilkan kata penting yang ada di basis data program.

7. Mendefinisikan List Kata Penting

Kata-kata penting didefinisikan sebagai *pattern* yang akan dicari dari teks masukan pengguna. Kata penting yang digunakan sesuai dengan fitur-fitur yang tersedia pada bot.

8. Menampilkan Pesan Error

Fitur ini memanfaatkan *string matching* untuk mencari suatu *pattern* dari teks masukan pengguna. Jika tidak ditemukan satupun *pattern* yang cocok maka bot akan menyimpulkan bahwa teks yang diberikan pengguna bukanlah sebuah permintaan yang valid, sehingga bot akan menampilkan pesan error.

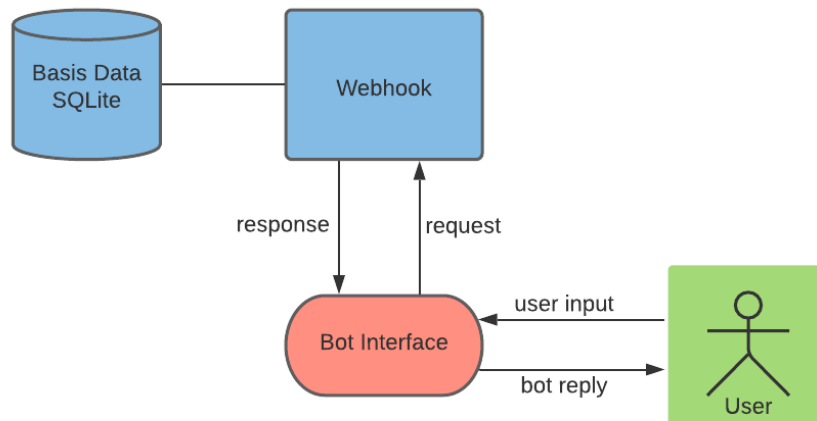
9. Memberikan Rekomendasi Kesalahan Kata

Fitur ini akan memeriksa kecocokan setiap kata yang diberikan pengguna dengan semua kata penting yang terdefinisi pada bot. Nilai kecocokan antara dua kata dilakukan dengan menggunakan algoritma levenshtein yang mana merupakan algoritma dengan pendekatan *dynamic programming*.

B. Fitur Fungsional dan Arsitektur *Chatbot*

Pada chatbot yang kami buat, terdapat 9 fitur utama seperti yang sudah dijelaskan pada bagian sebelumnya, yaitu

1. Menambahkan *Task* Baru
2. Melihat Daftar *Task* yang Harus Dikerjakan
3. Menampilkan *Deadline* dari Suatu *Task* Tertentu
4. Memperbaharui *Task* Tertentu
5. Menandai Bahwa Suatu *Task* Sudah Selesai Dikerjakan
6. Menampilkan Opsi *Help* yang Difasilitasi oleh *Assistant*
7. Mendefinisikan List Kata Penting
8. Menampilkan Pesan Error
9. Memberikan Rekomendasi Kesalahan Kata



Arsitektur chatbot yang kami buat secara sederhana dapat dilihat pada gambar di atas. Pertama-tama, user akan mengirimkan text kepada antarmuka *chatbot*. Kemudian text pengguna akan dikirimkan ke *webhook* sebagai *request*. Setelah *request* sampai ke *webhook*, text yang diberikan pengguna akan diproses secara logik dengan algoritma-algoritma yang sudah diimplementasikan. Pada proses ini juga terjadi berbagai transaksi dengan basis data. Setelah diproses, *webhook* akan mengirimkan *response* yang sesuai tergantung dari permintaan yang diberikan. Kemudian *response* akan ditampilkan berupa balasan dari *chatbot*.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

A. Spesifikasi Teknis Program

Aplikasi chatbot yang kami buat terdiri atas beberapa modul, diantaranya yaitu modul `app.py`, `basisdata.py`, `feature.py`, `matcher.py`, `addTugas.py`. Modul `app.py` berisi program utama web dimana kami menggunakan framework Flask. Modul ini terdiri dari dua fungsi yaitu `home()` untuk *rendering* halaman utama dan `get_bot_response()` untuk menentukan respons apa yang tepat untuk diberikan oleh bot.

Modul `basisdata.py` adalah modul yang berisi berbagai fungsi-fungsi dan prosedur untuk koneksi dengan basis data dimana kami menggunakan SQLite untuk basis datanya. Fungsi-fungsi pada modul ini yaitu `createDB()`, `clearDB()`, `update_daftar_tugas()`, `getlist_daftar_tugas()`, `insert_fitur()`, `getlist_fitur()`, dan fungsi-fungsi lainnya yang terkait dengan basis data.

Modul `feature.py` adalah modul yang berisi berbagai fitur yang disediakan oleh bot. Pada modul ini terdapat fungsi `process()` untuk memproses text yang diinput pengguna. Selain itu, pada modul ini juga terdapat fungsi `tampilDeadline()`, `tandaiTask()` dan `help()`.

Modul `matcher.py` adalah modul yang berisi algoritma pencocokan string. Modul ini terdiri dari fungsi `booyer_moore()` yang merupakan algoritma pencocokan string dengan Booyer Moore. Kemudian pada modul ini juga terdapat fungsi `levenshtein()` yang digunakan untuk menghitung kemiripan dua buah kata dengan algoritma Levenshtein yang mana merupakan algoritma dengan pendekatan *dynamic programming*.

B. Penjelasan Tata Cara Penggunaan Program

1. Menambahkan task baru

Rumus: `<kata penting> <nama/kode matkul> <topik> pada/tanggal <tanggal>`

Contoh:

- a. Tubes IF2211 Chatbot pada 29 april 2021
- b. tucil IF2211 makalah penerapan tanggal 2/05/2021
- c. Pr basdat bab 5 sampai 7 pada 15 mei 2021
- d. Bot ingatkan ujian IF2211 bab 5 -10 tanggal 10 mei 2021

2. Melihat task yang harus dikerjakan

a. Seluruh task yang sudah tercatat oleh asisten

Kata kunci yang bisa digunakan dalam menampilkan semua task yang telah dicatat adalah kata “deadline” dan “sejauh” .

Rumus: **deadline** <**sejauh**>

Contoh:

1. **Deadline** **sejauh** ini ada apa saja?
2. Bot, **deadline** **sejauh** ini ada apa saja?

b. Berdasarkan periode tertentu

1. Pada periode tertentu

Kata kunci yang bisa digunakan adalah kata “antara” yang diikuti tanggal kemudian kata “sampai” yang diikuti tanggal pula.

Rumus: **antara** <**tanggal**> **sampai** <**tanggal**>

Contoh:

- a. **Antara** **28/04/2021** **sampai** **29/05/2021**
- b. Deadline **antara** **28/04/2021** **sampai** **29/05/2021** apa saja?
- c. Bot ada deadline **antara** **28/04/2021** **sampai** **29/05/2021** apa aja?

2. N minggu ke depan

Kata yang bisa digunakan dalam menampilkan N dimana menyatakan banyaknya hari diikuti kata “minggu” kemudian ada kata “depan” dalam memasukkan pengguna.

Rumus: <**N**> **minggu** **depan**

Contoh:

- a. Deadline **3** **minggu** ke **depan**
- b. Apa saja deadline **3** **minggu** ke **depan**?

3. N hari ke depan

Kata yang bisa digunakan dalam menampilkan N dimana menyatakan banyaknya hari diikuti kata “hari” kemudian ada kata “depan” dalam memasukkan pengguna.

Rumus: <**N**> **hari** **depan**

Contoh:

- a. Deadline **3** **hari** ke **depan**
- b. Apa saja deadline **3** **hari** ke **depan**?

4. Hari ini

Kata kunci yang bisa digunakan adalah kata “deadline” diikuti kata “hari”.

Rumus: **deadline hari**

Contoh:

- a. **Deadline hari** ini ada apa saja?
- b. Apa saja **deadline hari** ini?

c. Berdasarkan jenis kata penting

Pada dasarnya fitur ini merupakan pengembangan lebih lanjut dari fitur sebelumnya sehingga kata kunci tinggal menggunakan kata kunci jenis di setiap kata kunci sebelumnya. Untuk lebih jelas bisa dilihat contoh dibawah ini.

Rumus:

1. **<jenis> antara <tanggal> sampai <tanggal>**
2. **<jenis> <N> minggu depan**
3. **<jenis> <N> hari depan**
4. **<jenis> deadline hari**

Contoh:

1. **Tubes Antara 28/04/2021 sampai 29/05/2021**
2. **Deadline PR antara 28/04/2021 sampai 29/05/2021** apa saja?
3. **Bot ada deadline tucil antara 28/04/2021 sampai 29/05/2021** apa aja?
4. **Deadline PR 3 minggu ke depan**
5. **Apa saja deadline tubes 3 minggu kedepan?**
6. **Deadline tubes 3 hari ke depan**
7. **Apa saja deadline PR 3 hari kedepan?**
8. **Deadline tubes hari** ini ada apa saja?
9. **Apa saja deadline pr hari** ini?

3. Menampilkan *deadline* dari suatu *task* tertentu

Kata kunci yang harus dimasukkan jika ingin melihat *deadline* dari suatu *task* adalah kapan dan *when* sebagai parameter waktu. Kemudian masukkan jenis *task* yang ingin dicari. Jika pengguna hanya menuliskan tugas maka *chatbot* akan menampilkan sembarang jenis tugas dari suatu mata kuliah yang ingin diketahui. Jika pengguna menuliskan jenis tugas yang ingin ditampilkan, *chatbot* akan menampilkan *deadline* dari mata kuliah tersebut sesuai dengan jenis tugas yang diinginkan.

Contoh:

- Pesan dari pengguna:
kaplan deadline tucil if2230?
- Pesan dari *chatbot*:
2021-07-21

4. Memperbarui taks tertentu

Kata kunci yang harus dimasukkan adalah kata kunci ID dari task diikuti kata “undur” atau “ubah” yang diikuti kata “pada” dan kemudian keterangan tanggal. Apabila dirumuskan menjadi <ID> <undur/ubah> pada <tanggal>.

Contoh:

- Pesan dari pengguna:
ID 4 diundur pada 03 mei 2021
- Pesan dari *chatbot*:
Deadline Tugas ID 4
berhasil diperbarui menjadi 2021-05-03
- Pesan dari pengguna:
ID 4 diubah pada 02 mei 2021
- Pesan dari *chatbot*:
Deadline Tugas ID 4
berhasil diperbarui menjadi 2021-05-02

5. Menandai bahwa suatu task selesai dikerjakan

Kata kunci yang harus dimasukkan jika ingin menandai suatu *task* adalah *done* dan selesai. Kemudian masukkan ID dari *task* yang ingin ditandai selesai.

Contoh:

- Pesan dari pengguna:
saya selesai mengerjakan task 1
- Pesan dari *chatbot*:
Task ditandai selesai

6. Menampilkan opsi help yang difasilitasi oleh asisten

Kata kunci yang harus dimasukkan jika ingin menampilkan opsi *help* adalah *help*, bisa, dan lakukan. Kemudian fitur dan kata penting akan ditampilkan.

Contoh:

- Pesan dari pengguna:

apa yang bisa dilakukan bot ini?

- Pesan dari *chatbot*:

[Fitur]

1. Menambahkan task baru

2. Melihat daftar task

3. Menampilkan deadline dari suatu task tertentu

4. Memperbaharui task tertentu

5. Menandai bahwa suatu task sudah dikerjakan

6. Memberikan rekomendasi kata

[Daftar kata penting]

1. deadline

2. tubes

3. tucil

4. kuis

5. ujian

6. pr

7. Menampilkan pesan error jika assistant tidak dapat mengenali masukan user

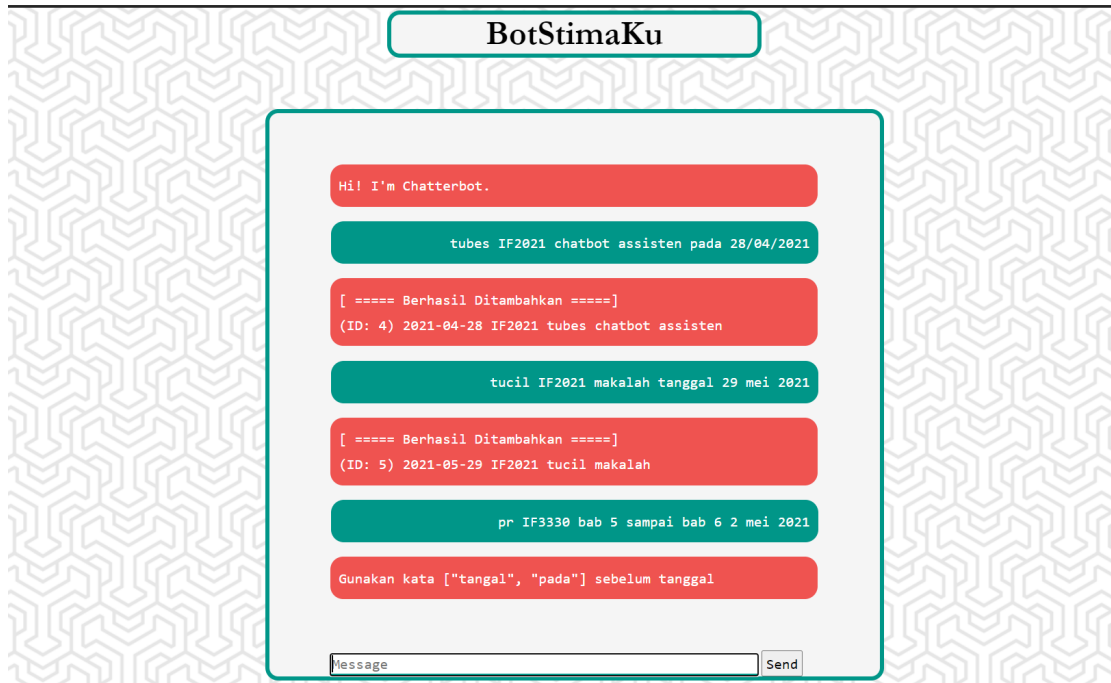
Jika bot tidak mendeteksi satupun kata penting yang terdefinisi ada pada masukan, maka bot akan menampilkan pesan error.

8. Chatbot dapat memberikan rekomendasi kata jika terdapat kesalahan kata pada perintah yang dituliskan pengguna.

Misalnya jika pengguna memasukkan kalimat ‘dedline sejah ini apa saja?’, maka bot akan memberikan rekomendasi ‘deadline sejauh ini apa saja?’.

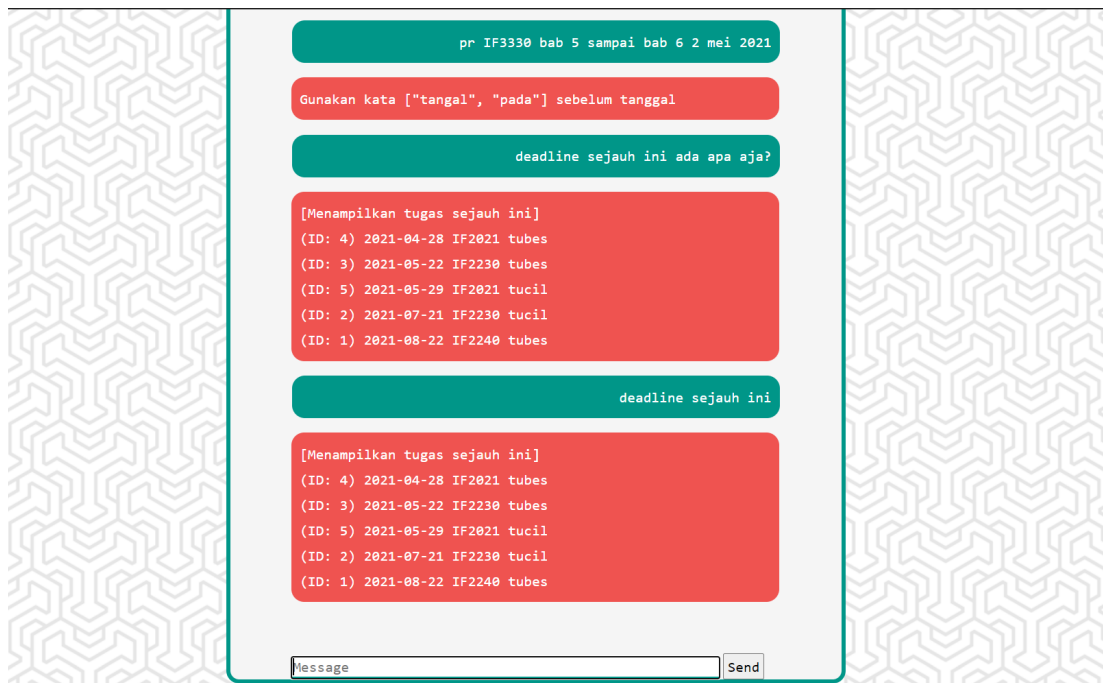
C. Hasil Pengujian

1. Menambahkan task baru



2. Melihat task yang harus dikerjakan

a. Seluruh task yang sudah tercatat oleh asisten



- b. Berdasarkan periode tertentu
1. Pada periode tertentu

The screenshot shows a chat window with a light gray background and a white border. The chat area contains several red and teal message bubbles. The red bubbles contain task details, and the teal bubbles contain deadline information. At the bottom, there is a text input field labeled "Message" and a "Send" button.

[Menampilkan deadline antara 2021-04-28 - 2021-05-20]
(ID: 4) 2021-04-28 IF2021 tubes

deadline antara 28/04/2021 sampai 20/05/2021

[Menampilkan deadline antara 2021-04-28 - 2021-05-20]
(ID: 4) 2021-04-28 IF2021 tubes

deadline antara 28/04/2021 sampai 29/05/2021

[Menampilkan deadline antara 2021-04-28 - 2021-05-29]
(ID: 4) 2021-04-28 IF2021 tubes
(ID: 3) 2021-05-22 IF2230 tubes
(ID: 5) 2021-05-29 IF2021 tucil

deadline antara 28/04/2021 sampai 20/08/2021

[Menampilkan deadline antara 2021-04-28 - 2021-08-20]
(ID: 4) 2021-04-28 IF2021 tubes
(ID: 3) 2021-05-22 IF2230 tubes
(ID: 5) 2021-05-29 IF2021 tucil
(ID: 2) 2021-07-21 IF2230 tucil

Message Send

2. N minggu ke depan

The screenshot shows a chat window with a light gray background and a white border. The chat area contains several red and teal message bubbles. The red bubbles contain task details, and the teal bubbles contain specific deadline information. At the bottom, there is a text input field labeled "Message" and a "Send" button.

(ID: 4) 2021-04-28 IF2021 tubes
(ID: 3) 2021-05-22 IF2230 tubes
(ID: 5) 2021-05-29 IF2021 tucil
(ID: 2) 2021-07-21 IF2230 tucil

deadline 3 minggu ke depan ?

[Menampilkan Tugas 3 minggu ke depan]
(ID: 4) 2021-04-28 IF2021 tubes

deadline 4 minggu kedepan ada apa aja?

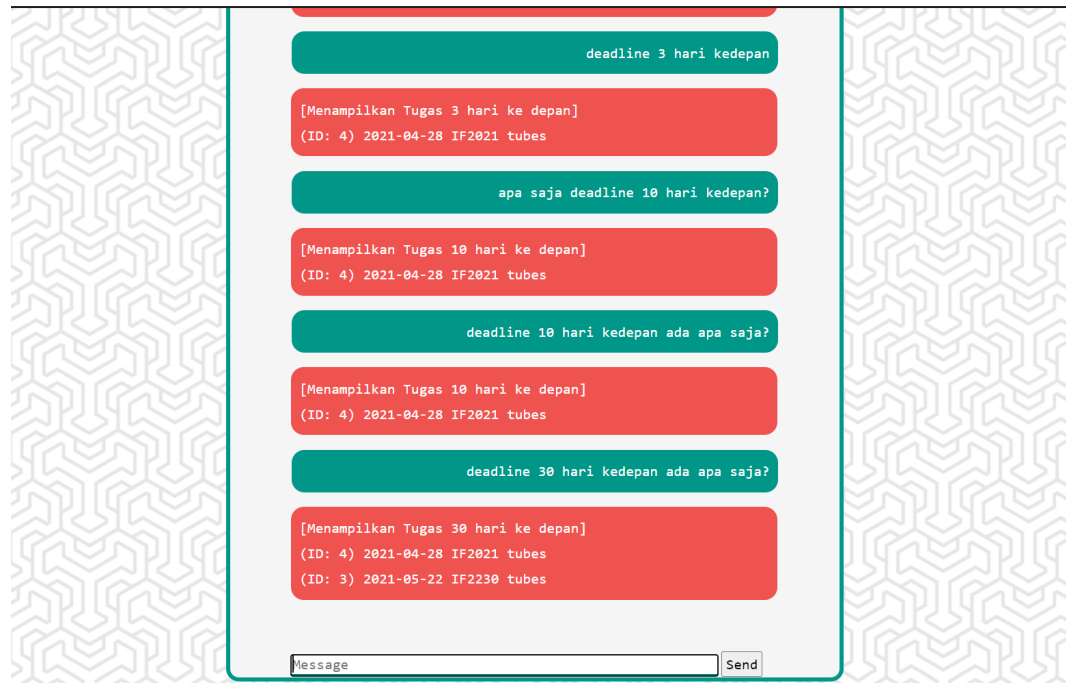
[Menampilkan Tugas 4 minggu ke depan]
(ID: 4) 2021-04-28 IF2021 tubes
(ID: 3) 2021-05-22 IF2230 tubes

deadline 10 minggu kedepan ada apa saja?

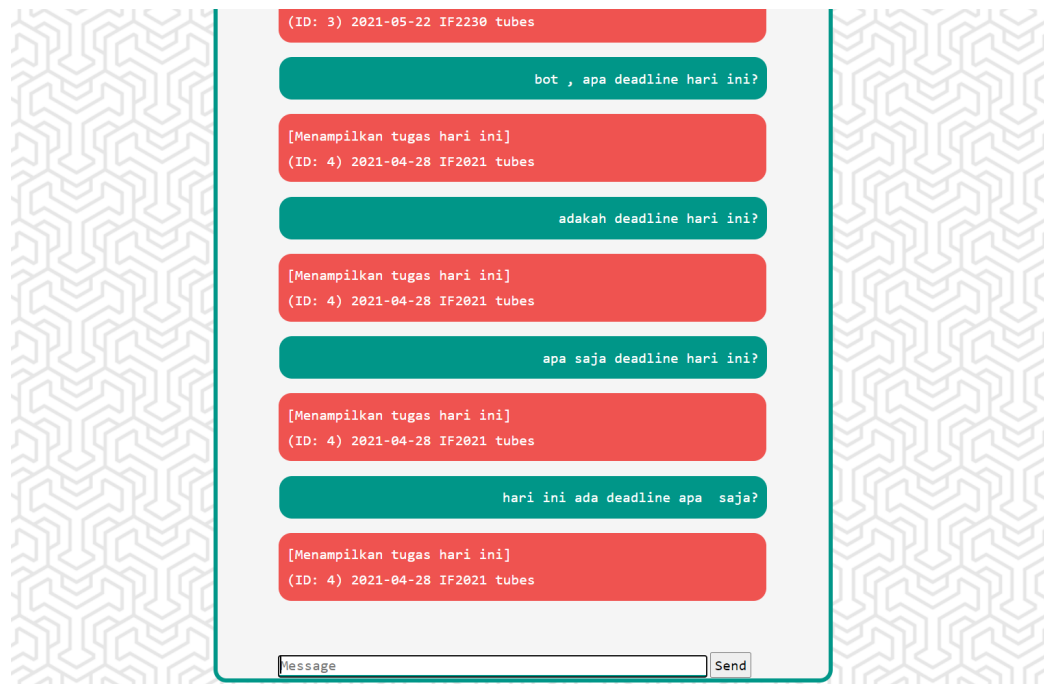
[Menampilkan Tugas 10 minggu ke depan]
(ID: 4) 2021-04-28 IF2021 tubes
(ID: 3) 2021-05-22 IF2230 tubes
(ID: 5) 2021-05-29 IF2021 tucil

Message Send

3. N hari ke depan



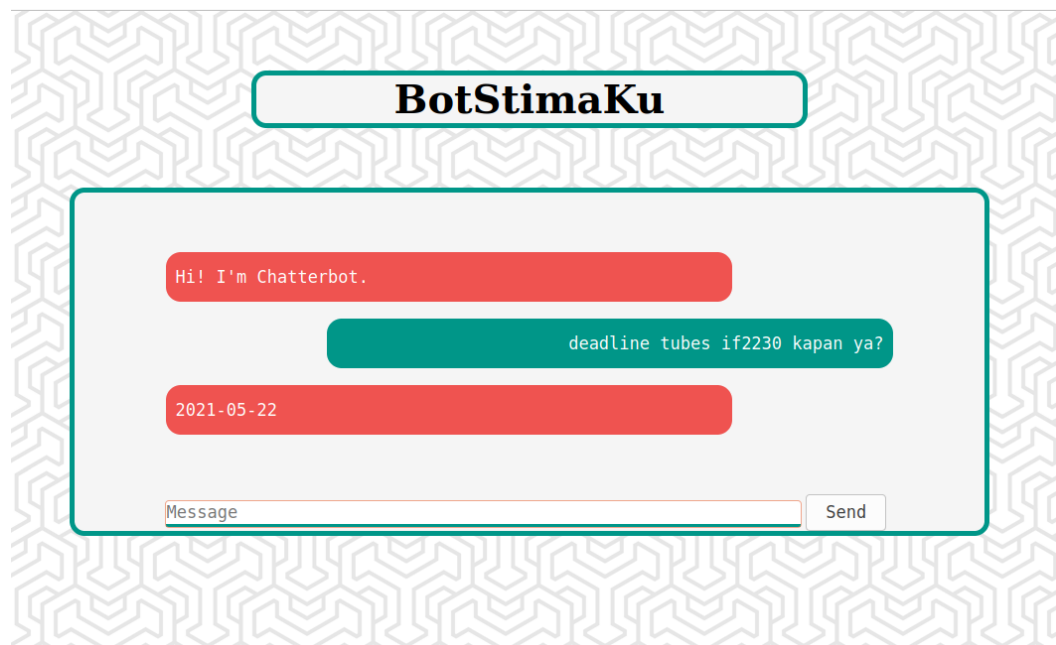
4. Hari ini



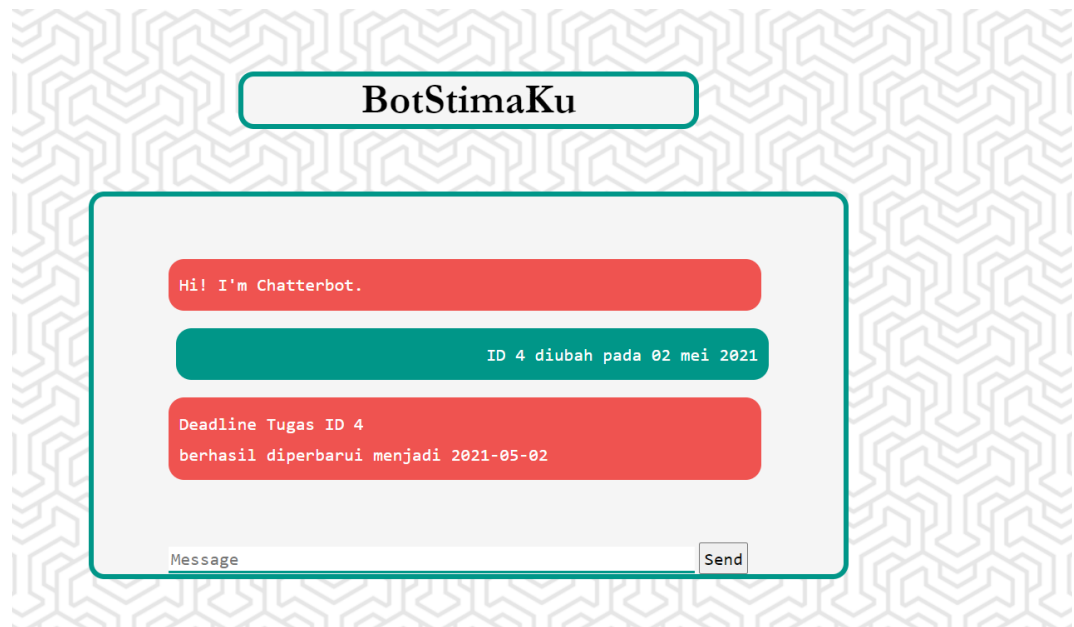
c. Berdasarkan jenis kata penting



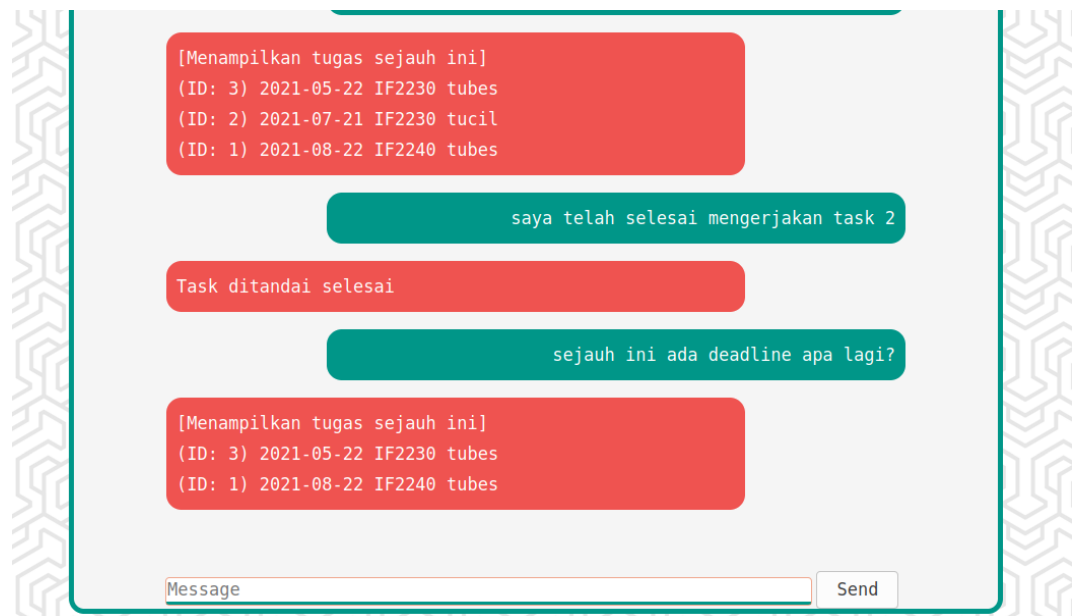
3. Menampilkan *deadline* dari suatu *task* tertentu



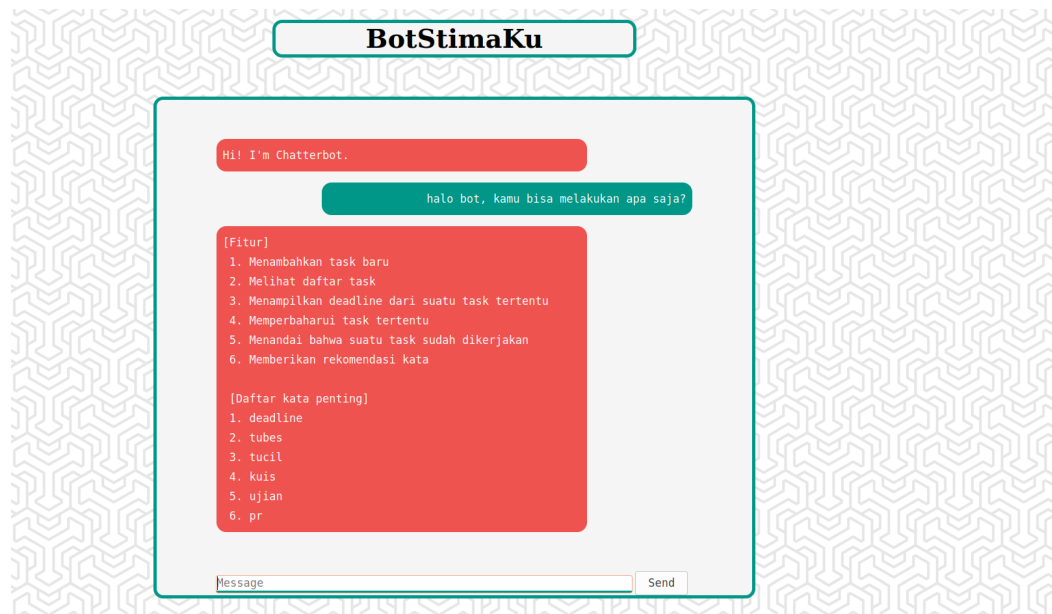
4. Memperbarui taks tertentu



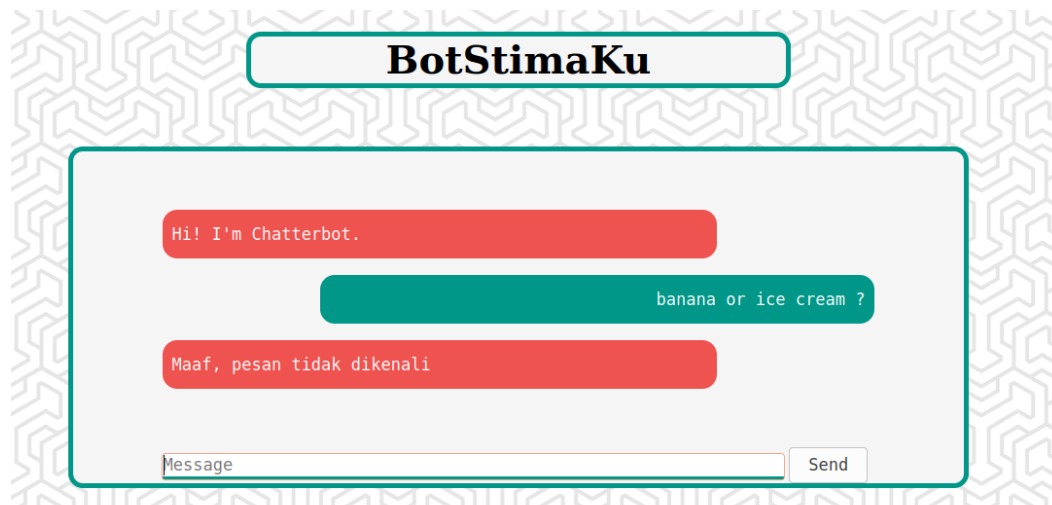
5. Menandai bahwa suatu task selesai dikerjakan



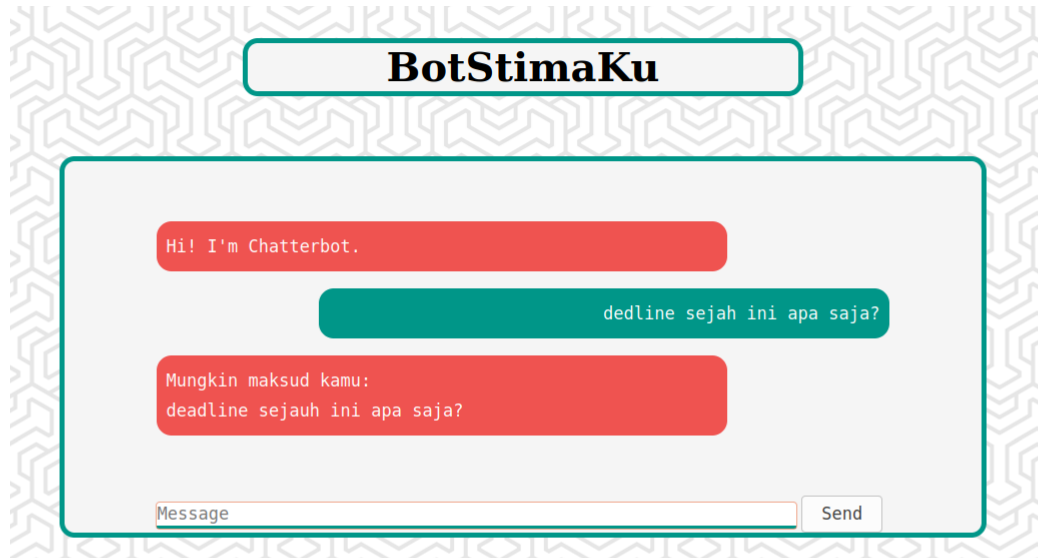
6. Menampilkan opsi help yang difasilitasi oleh asisten



7. Menampilkan pesan error jika assistant tidak dapat mengenali masukan user



8. Chatbot dapat memberikan rekomendasi kata jika terdapat kesalahan kata pada perintah yang dituliskan pengguna



D. Analisis Hasil Pengujian

Algoritma yang digunakan sebagai pencocokan *string* dari masing-masing fitur adalah algoritma Boyer-Moore dan *Regular Expression*, sementara untuk fitur rekomendasi kata menggunakan algoritma Levenshtein. Algoritma Boyer-Moore dan Levenshtein diimplementasikan dengan membuat sebuah fungsi yang direalisasikan sendiri sedangkan *regular expression* menggunakan *library* yang telah disediakan bahasa pemrograman Python. Berdasarkan hasil pengujian yang dilakukan *Regular Expression* lebih baik digunakan dalam melakukan *string matching* dengan angka atau simbol-simbol yang menimbulkan keambiguan saat dibandingkan (misalnya angka 12 dapat dibaca sebagai angka 1 apabila menggunakan algoritma Boyer-Moore sedangkan *Regular Expression* akan membaca angka tersebut dengan tepat).

Walaupun demikian dengan algoritma Boyer-Moore masih bisa digunakan dengan baik dalam mengembangkan fitur-fitur yang ada dalam tugas ini. Algoritma Boyer-Moore banyak digunakan dalam pengecekan kata yang mungkin bisa dilakukan dengan algoritma ini dan selebihnya menggunakan *Regular Expression* ataupun manipulasi string yang telah ada di Python. Di sisi yang lain, dengan menggunakan algoritma Levenshtein, kecocokan antara dua buah kata dapat ditentukan yaitu banyaknya huruf yang berbeda per banyaknya huruf pada kata.

Algoritma Levenshtein dapat menentukan perbedaan kata dengan tepat namun kekurangannya membutuhkan waktu yang sangat lama apabila bertemu dengan kata yang panjang.

Secara keseluruhan, berdasarkan hasil pengujian penerapan string matching dan *regular expression* dalam membangun *deadline reminder assistant* dapat berjalan dengan baik dan sesuai harapan.

BAB V

KESIMPULAN, SARAN, DAN KOMENTAR/REFLEKSI

A. Kesimpulan

Setelah menyelesaikan tugas besar 3 ini, didapatkan bahwa algoritma *string matching* dan *regular expression* dapat bisa diterapkan untuk membuat aplikasi *chatbot* dengan membekali *chatbot* dengan kata-kata penting yang dicocokkan dengan pesan dari pengguna. Algoritma *string matching* dengan menggunakan *regular expression*, Boyer Moore, atau Knuth-Morris-Pratt efektif untuk melakukan pencocokan kata terbukti dengan respon yang diberikan *chatbot* tidak memakan waktu yang lama.

B. Saran

Pada pengerjaan tugas besar ini, kelompok kami masih memiliki kekurangannya. Tugas besar 3 ini diharapkan dapat menjadi referensi mahasiswa dalam pengerjaan tugas atau saat sudah terjun dalam pengembangan aplikasi berbasis *website* dan sebagai langkah awal dalam pembelajaran *machine learning*.

C. Komentar

Menurut kelompok kami, tugas besar 3 yang diberikan ini sangat menarik. Tugas besar ini dapat menambah wawasan mahasiswa dalam pengembangan aplikasi berbasis *website*. Aplikasi *chatbot* biasanya ditemukan dalam bentuk aplikasi pesan seperti Whatsapp dan LINE yang pada dasarnya adalah aplikasi berbasis *mobile*. Namun, tugas kali ini dibuat cukup unik, yaitu dengan membuat aplikasi *chatbot* di *website*. Secara keseluruhan tugas besar ini sangat menantang dan meningkat pola pikir serta kreativitas terutama dalam mendesain antarmuka sebuah program.

D. Refleksi

Pada proses pengerjaan tugas ini, kelompok mengalami beberapa kendala teknis dan beberapa kendala non-teknis. Namun kendala sudah dapat terselesaikan. Kendala teknis berupa, mulanya program masih sering *error* dan terbatasnya kreativitas dalam mendesain antarmuka program. Sedangkan kendala non-teknis berupa *time management* kelompok yang kurang baik, sehingga pengerjaan tugas ini sempat tertunda akibat banyaknya tugas lain yang batas pengumpulannya lebih cepat. Kelompok kami berharap, kami mampu memperbaiki *skill time management*.

REFERENSI

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Tugas-Besar-3-IF2211-Strategi-Algoritma-2021.pdf> (diakses pada 12 April 2021).

<https://github.com/chamkank/flask-chatterbot> (diakses pada 25 April 2021).

<https://dev.to/lordofdexterity/deploying-flask-app-on-heroku-using-github-50nh> (diakses pada 25 April 2021).

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf> (diakses pada 27 April 2021).

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Modul-Praktikum-NLP-Regex.pdf> (diakses pada 27 April 2021).

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf> (diakses pada 27 April 2021).

<https://idcloudhost.com/mengenal-apa-itu-chatbot-definisi-cara-kerja-fungsinya/> (diakses pada 27 April 2021).