

TUGAS BESAR III

Algoritma String Matching dan Regular Expression

Pada Deadline Reminder Assistant

IF2211 – Strategi Algoritma



Oleh

Kelompok 70 : Gespra

Ariya Adinatha 13519048

Moses Ananta 13519076

Jonathan Richard Sugandhi 13519128

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI 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
IF2211 Strategi Algoritma - Tugas Besar 3 2

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.

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

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
7. Mendefinisikan list kata penting terkait apakah itu merupakan suatu task atau tidak
 - a. Minimal terdapat 5 kata penting berbeda, contohnya adalah: ["Kuis", "Ujian", "Tucil", "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
 - a. Berikan rekomendasi kata jika perintah masukan pengguna mismatch dengan daftar kata yang diterima chatbot, namun masih memiliki tingkat kemiripan di atas 75%.
 - b. Contoh interaksi
 - c. Ada berbagai matriks yang dapat dimanfaatkan untuk mencari kemiripan kata, salah satunya adalah Levenshtein distance yang diukur melalui pendekatan dynamic programming. Anda dapat mempelajari Levenshtein distance melalui pranala ini.

BAB II

LANDASAN TEORI

2.1. Algoritma Knuth Morris Pratt

Algoritma *Knuth Morris Pratt* merupakan salah satu algoritma yang digunakan untuk mencari *string*, dikembangkan oleh Donald E. Knuth, James H. Morris, dan Vaughan R. Pratt pada tahun 1966. Algoritma ini merupakan jenis *Exact String Matching Algorithm* yang merupakan pencocokan *string* secara tepat berdasarkan susunan karakter maupun jumlah karakter dari *string* yang sama. Pada algoritma *Knuth Morris Pratt* menyimpan informasi yang digunakan dalam melakukan pergeseran lebih jauh, tidak hanya satu karakter seperti algoritma *Brute Force*.

2.2. Algoritma Boyer Moore

Algoritma *Boyer Moore* merupakan salah satu algoritma pencarian *string* yang dipublikasikan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum, tidak seperti algoritma pencarian *string* yang ditemukan sebelumnya, algoritma *Boyer Moore* mulai mencocokkan karakter dari sebelah kanan dari sebuah *pattern*.

2.3. Algoritma Regular Expression

Regular Expression atau disingkat menjadi regex merupakan notasi standar yang mendeskripsikan suatu pola berupa urutan *string* atau serangkaian karakter yang mendefinisikan sebuah pola pencarian. Pola tersebut biasanya digunakan oleh algoritma pencarian *string* untuk melakukan operasi pencarian pada *string*.

2.4. Chatbot

Chatbot merupakan sebuah program komputer yang dapat mensimulasikan percakapan layaknya manusia. Teknologi ini juga dikenal sebagai asisten digital yang dapat memahami serta memproses permintaan pengguna dan memberikan jawaban yang relevan dengan cepat. Sebutan “bot” berasal dari singkatan robot. Dimana bot itu sendiri dapat menirukan percakapan manusia dalam bentuk text maupun suara yang diaplikasikan terhadap website dan juga aplikasi.

BAB III

ANALISIS PEMECAHAN MASALAH

3.1. Langkah Penyelesaian

Setelah web berhasil dibangun , maka pengguna akan memasukan teks input pada kolom input yang sudah disediakan. Setelah pengguna memasukan input teks tersebut maka program memproses masukan teks pengguna berdasarkan kata kunci dan kata penting yang ditemukan pada teks masukan pengguna. Setiap kata kunci dan kata penting pada program ini dicek menggunakan Algoritma Boyer Moore terhadap masukan teks pengguna. Dari kata kunci dan kata penting yang sudah ditemukan inilah yang menentukan fitur dan responsi bot yang akan diberikan kepada pengguna.

Terdapat daftar kata kunci yang terdefinisi di program yaitu "help", "assistant", "diundur", "selesai", "kapan", "deadline", "tugas", "apa saja", dan "apa aja". Jika program tidak menemukan kata kunci pada teks masukan pengguna , maka program akan menganggap bahwa pengguna ingin memasukan task baru. Program akan mengecek apakah pada teks masukan pengguna terdapat tanggal dan, kode kuliah yang didapatkan dengan menggunakan regex, juga kata penting yang terdiri dari kata "Kuis", "Ujian", "Tucil", "Tubes", dan "Praktikum" . Jika tidak didapati salah satu dari ketiga unsur tersebut maka bot akan memberikan respon pesan kesalahan. Sebaliknya jika didapati ketiga unsur tersebut maka bot akan memberikan respon berupa data task yang sudah diberikan pengguna dan program akan menambahkan task tersebut pada database.

Jika ditemukan kata kunci "deadline" atau "tugas" pada teks masukan pengguna, pertama-tama program akan mencari kata penting yang ada pada masukan pengguna. Jika terdapat kata penting dalam masukan pengguna, maka kata penting akan dicek. Apabila terdapat kata penting yang berkategori "tidak memiliki deadline", maka akan dihasilkan pesan error. Jika kata penting ada dan sesuai, maka hasil akan di filter sesuai masukan pengguna, jika tidak ada kata penting, maka akan ditampilkan semua task yang memiliki deadline. Setelah mencari kata penting, akan dicari keterangan waktu dan kode kuliah. Jika ditemukan keterangan waktu, maka hasil yang ditampilkan akan di-filter berdasarkan keterangan waktu tersebut, sama halnya dengan kode kuliah, apabila ditemukan maka akan di-filter sesuai dengan kode tersebut. Jika tidak ditemukan kode kuliah atau keterangan waktu, maka hasil yang ditampilkan akan di-filter.

Jika ditemukan kata kunci "kapan" pada teks masukan pengguna, maka terlebih dahulu program akan mencari kata penting masukan pengguna. Apabila tidak ditemukan, maka tidak dilakukan filtering. Lalu program akan mencari kode kuliah, jika ditemukan maka program akan mem-filter hasil. Lalu jika hasil yang didapat tidak ada akan diberi pesan "Tidak Ada", jika hasil yang didapat hanya 1 maka akan ditampilkan waktunya langsung, dan jika hasil yang didapat lebih dari 1 maka akan ditampilkan seluruh detail dari hasil.

Jika ditemukan kata kunci “apa saja” atau “apa aja” pada teks masukan pengguna, program akan bertindak seperti deadline namun menerima seluruh tipe task baik yang memiliki deadline maupun tidak.

Jika ditemukan kata kunci “diundur” pada teks masukan pengguna, program akan mencari id task dan tanggal dengan menggunakan regex. Jika ditemukan id task dan tanggal tersebut maka program akan mengubah tanggal pada id task yang bersangkutan pada database sesuai dengan tanggal yang diberikan pengguna dan bot akan memberikan pesan berhasil. Jika tidak ditemukan, maka bot akan memberikan pesan error.

Jika ditemukan kata kunci “selesai” pada teks masukan pengguna, program akan mencari id task dengan menggunakan regex. Jika ditemukan id task tersebut maka program akan mengubah status task menjadi True pada id task yang bersangkutan pada database dan bot akan memberikan pesan berhasil. Jika tidak ditemukan, maka bot akan memberikan pesan error.

Jika ditemukan kata kunci “help” atau “assistant” pada teks masukan pengguna, maka bot akan menampilkan daftar fitur dan daftar kata penting kepada pengguna.

3.2. Fitur Fungsional dan Arsitektur Chatbot yang dibangun

Dalam aplikasi Chatbot ini terdapat beberapa fitur fungsional yang dibangun, yaitu :

1. Menetapkan 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

Adapun Frontend yang digunakan untuk membangun aplikasi berbasis web ini melibatkan penggunaan html, css, dan javascript. Untuk Backend digunakan python dengan bantuan framework Flask dengan manajemen basis data dengan melibatkan bantuan json.

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Spesifikasi Teknis Program

a. Struktur Data

Data yang berupa task pada program ini disimpan dalam Tasks.db dengan urutan tanggal deadline task, kode kuliah,kategori task,deskripsi task,dan status pengerjaan task dalam format {"dd/mm/yyyy",<kode_kuliah>,<kategori_task>,<deskripsi_task>,boolean}.

b. Fungsi

- getBotChat() : digunakan untuk menerima respons dari chat pengguna
- getBotResponse() : digunakan untuk mengolah chat dari pengguna dengan memanggil fungsi handleInput()
- botChat() : template HTML dari bot chat
- userChar() : template HTML dari user chat
- getTime() : mendapatkan waktu dalam format HH:MM
- index() : menampilkan html
- booyer_moore_string_matching(sentence, matcher) : melakukan pencarian *string* dengan menggunakan algoritma *Boyer Moore*
- loadTask() : membaca database
- saveTask() : menyimpan database
- getListOfKataPenting() : mengembalikan kata penting yang terdaftar
- getFlag() : mengembalikan flag / kata kunci yang terdaftar
- GetInputDate(inputTask) : menerima tanggal
- DateConverter(inputListDate) : mengubah bentuk tanggal yang dimasukkan
- GetInputKodeKuliah(inputTask) : menerima input kode kuliah dan mengembalikan kode kuliah
- GetInputDeskripsiTugas(inputTask) : menerima deskripsi tugas
- GetKategoriTugas(inputTask): mengembalikan kategori tugas yang ditentukan berdasarkan kata penting yang ditemukan pada inputTask
- query(database, queries, idx):
- task_viewer(sentence, indeks, database):
- HandleInput(inputTask): mengembalikan respon bot sesuai dengan fitur yang ada yang ditentukan berdasarkan kata kunci dan kata penting yang ditemukan pada inputTask

c. Prosedur

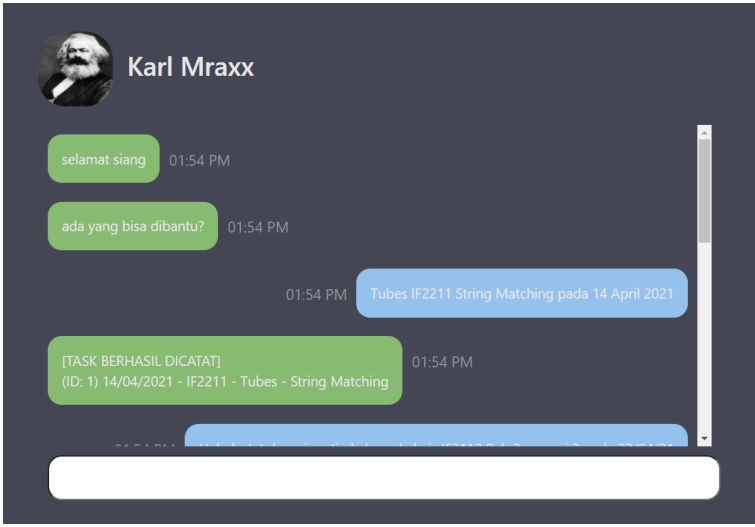
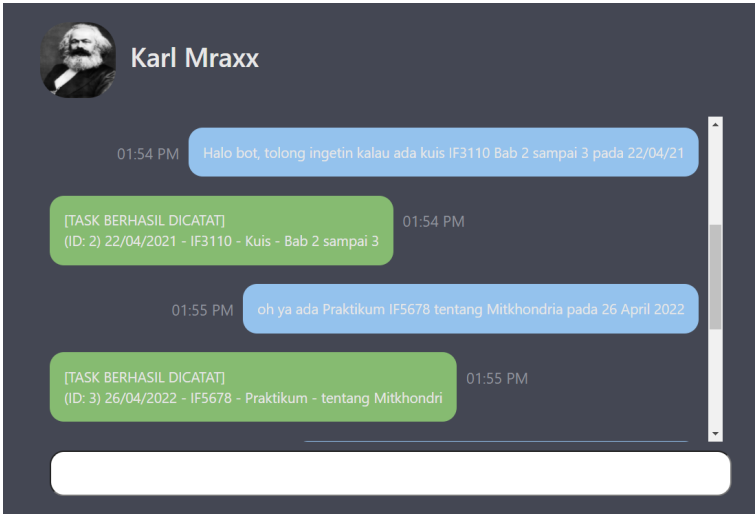
- Onload : saat pertama kali web app dijalankan, program akan mencari jam terkini dan mengucapkan salam berdasarkan waktu tersebut

4.2. Tata Cara Penggunaan Program

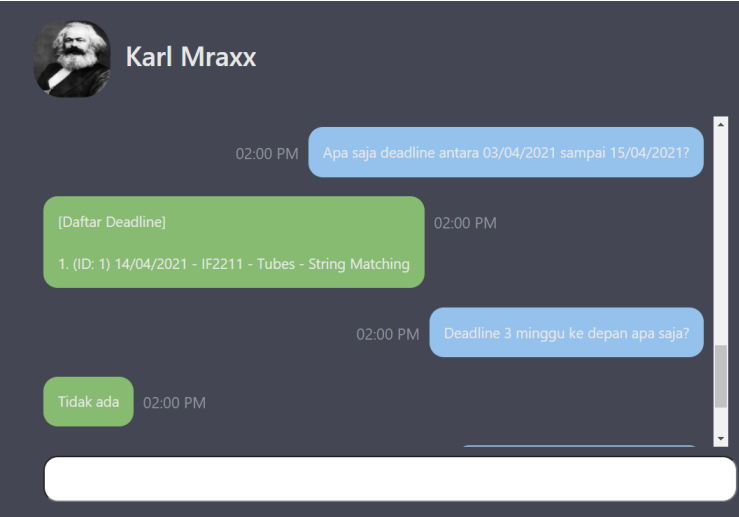
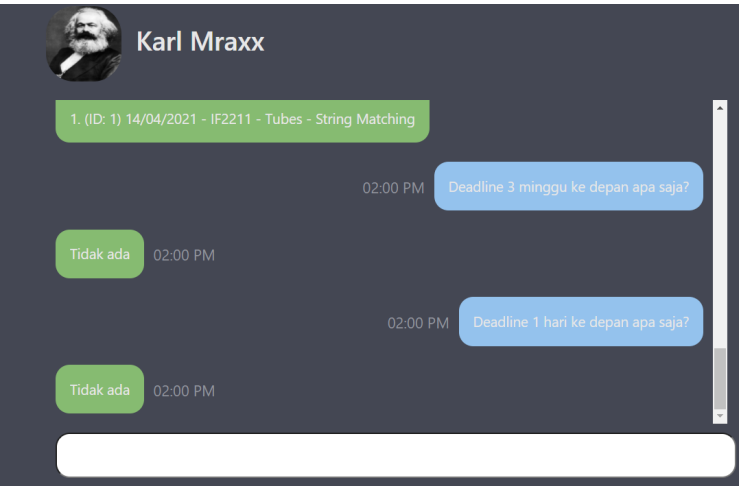
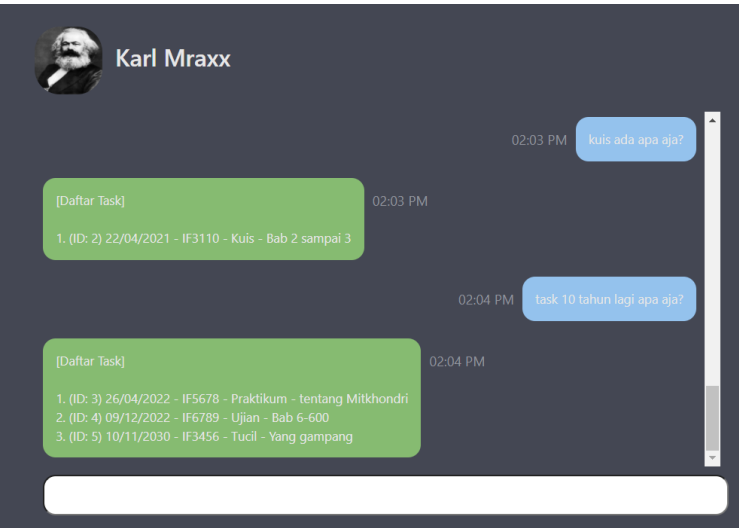
- Clone github https://github.com/ariyaadinatha/Tubes3_13519048.git
- Masuk ke folder hasil clone
- Buat virtual environment
- Aktifkan virtual environment
- Install Requirement dengan pip install
- Masuk ke dalam folder src
- Jalankan program app.py pada terminal dengan python

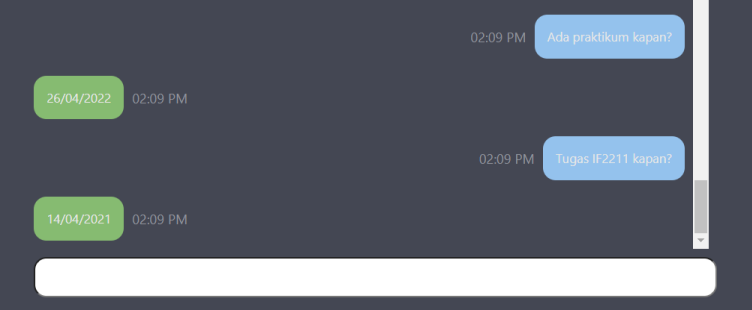
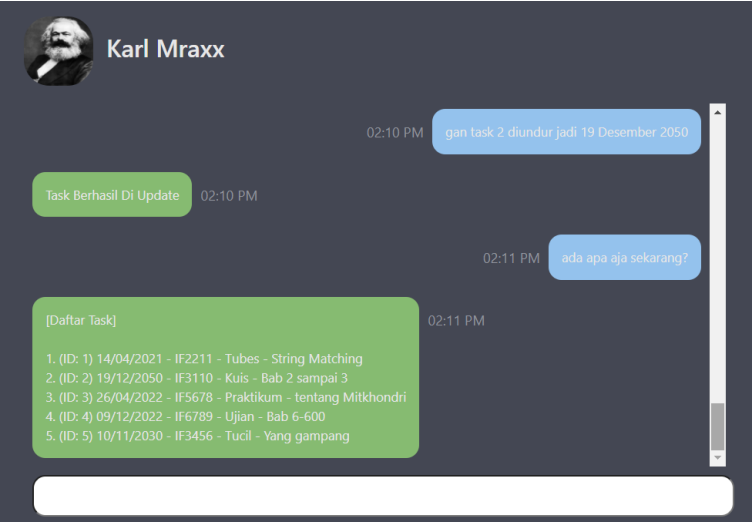
Ketika program sudah berjalan dengan python, akan muncul address web yang ketika dimasukan ke dalam web browser akan memunculkan tampilan web program. masukkan pesan yang diinginkan melalui text box yang ada.

4.3. Hasil Pengujian

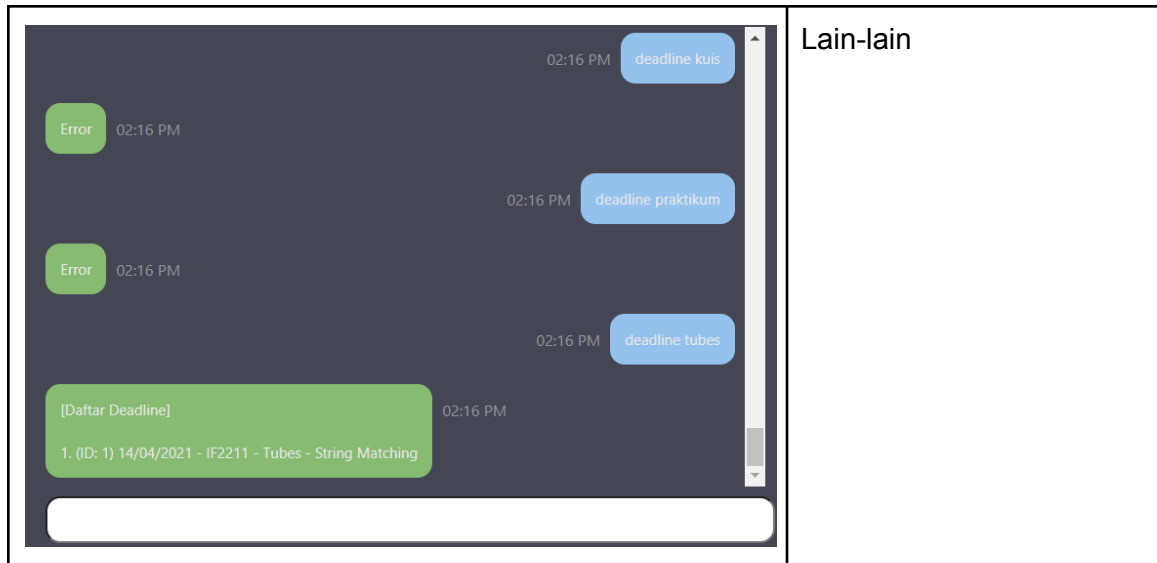
Gambar Hasil Pengujian	Keterangan
	Menguji fitur menambah task baru
	Menguji fitur menambah task baru

	<p>Menguji fitur menambah task baru</p>
	<p>Database hasil penambahan task di atas</p>
	<p>Melihat semua deadline yang ada (hanya tubes dan tupil yang punya deadline)</p>

	<p>Melihat semua deadline dengan keterangan waktu</p>
	<p>Melihat semua deadline dengan keterangan waktu</p>
	<p>Melihat suatu task secara spesifik dan melihat seluruh task dengan periode waktu</p>

	<p>Melihat waktu task secara spesifik</p>
	<p>Memperbarui tanggal deadline</p>
	<p>Menyatakan suatu tugas sudah selesai</p>

	<p>Menampilkan fitur dan kata penting</p>
	<p>Kasus input salah</p>



4.4. Hasil Analisis

Dari hasil pengujian tersebut, dapat dilihat aplikasi dapat menambahkan task baru sesuai dengan input-an pengguna. Aplikasi juga dapat melihat daftar task yang belum dikerjakan sesuai dengan keterangan waktu tertentu ataupun keterangan kode kuliah. Aplikasi juga dapat memperbaharui keterangan waktu dari setiap task yang sudah terdaftar. Aplikasi lalu dinilai juga dapat menandai suatu task telah selesai dikerjakan sehingga nantinya tidak akan tampil saat dilakukan pencarian task. Aplikasi juga dapat menampilkan seluruh fitur dari aplikasi dengan menuliskan keyword “help” atau “assistant”. Aplikasi juga dapat memberikan pesan kesalahan apabila pengguna menuliskan pesan yang tidak sesuai.

Dari hasil analisis diatas, dapat disimpulkan bahwa aplikasi dapat bekerja sebagaimana mestinya.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Pencocokan *string* dapat dilakukan dengan berbagai macam algoritma, diantaranya adalah algoritma *Knuth Morris Pratt*, algoritma *Boyer Moore*, dan juga algoritma *Regular Expression*. Dimana setiap algoritma ini memiliki pendekatannya masing masing dalam melakukan pencocokan *string*. Pada algoritma *Knuth Morris Pratt* akan dilakukan pendekatan pergeseran karakter, sedangkan untuk algoritma *Boyer Moore* akan melakukan pencocokan *string* dari kanan, lalu algoritma *Regular Expression* akan mencocokkan berdasarkan *pattern* dari *string*.

5.2. Saran

1. Mempelajari website development
2. Membaca dokumentasinya sampai mengerti
3. Menanyakan pertanyaan yang kurang jelas kepada asisten atau membaca FAQ

DAFTAR PUSTAKA

1. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf> (Diakses pada 27 April 2021; 21:05)
2. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regular-ex-2019.pdf> (Diakses pada 27 April 2021; 21:05)
3. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions (Diakses pada 28 April 2021; 00:46)
4. <https://www.javatpoint.com/daa-boyer-moore-algorithm> (Diakses pada 28 April 2021; 01:16)
5. <https://brilliant.org/wiki/knuth-morris-pratt-algorithm/> (Diakses pada 27 April 2021; 23:51)
6. https://www.niagahoster.co.id/blog/chatbot-adalah/#Apa_itu_Chatbot (Diakses pada 27 April 2021; 23:33)