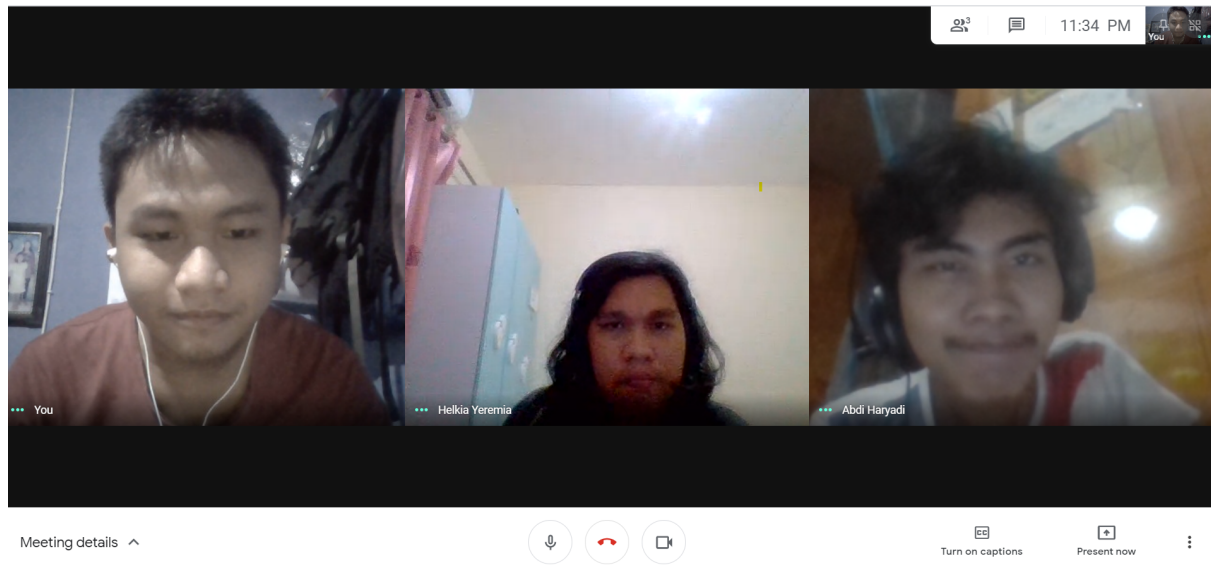


LAPORAN TUGAS BESAR 3

IF2211 STRATEGI ALGORITMA



Nama Kelompok : TimeTeam

Anggota Kelompok :

1. Mochammad Fatchur Rochman 13519009
2. Helkia Yermia 13519056
3. M. Abdi Haryadi. H 13519156

Tanggal Pembuatan : 28 April 2021

PROGRAM STUDI TEKNIK INFORMATIKA
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.

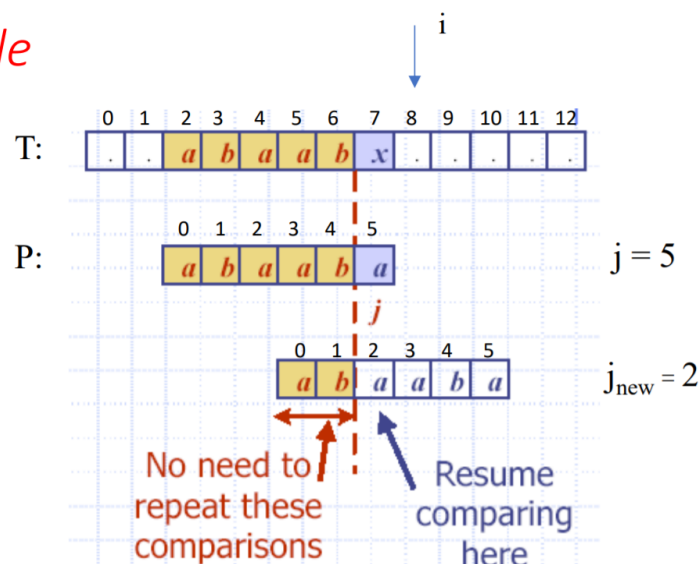
BAB II

LANDASAN TEORI

Topik utama tugas besar ketiga ini adalah *pattern matching*. Dalam pattern matching, akan ada sebuah text dan pattern. Pada umumnya text akan lebih panjang dari pattern. Persoalan yang harus diselesaikan adalah mencari kemunculan pertama dari pattern dalam sebuah text. Untuk menyelesaikan persoalan ini terdapat beberapa teknik seperti Knuth-Morris-Pratt (KMP) algorithm, The Boyer-Moore Algorithm, dan Regular expression.

Knuth-Morris-Pratt (KMP) algorithm merupakan salah satu algoritma pattern matching. Algoritma ini mencari pattern dalam sebuah text dengan urutan dari kiri ke kanan (mirip dengan algoritma brute force) namun pattern di shift dengan cara yang lebih “pintar” dibandingkan bruteforce. Ide utama dari algoritma ini adalah mengurangi shifting pattern untuk menghindari operasi perbandingan yang tidak dibutuhkan. Hal ini dapat dilakukan dengan melakukan shift pattern menjadi prefix terbesar yang sekaligus menjadi suffix.

Example



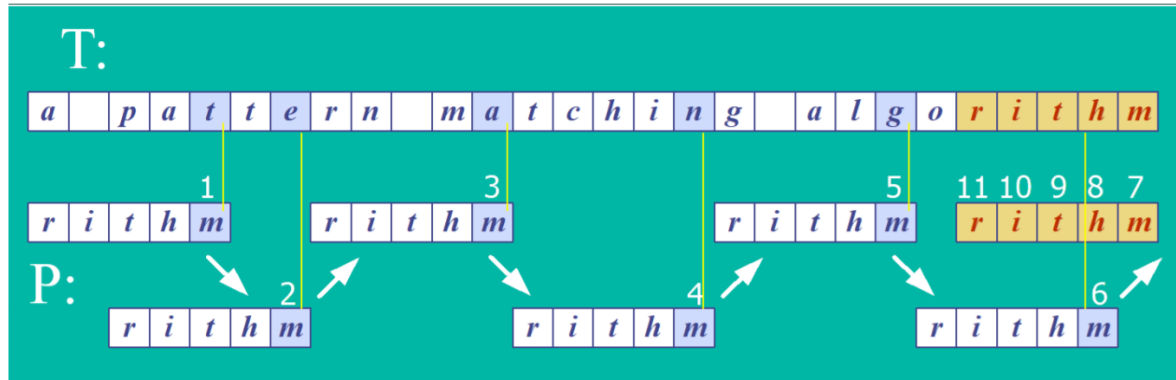
Kompleksitas waktu untuk menghitung fungsi pinggiran KMP adalah $O(m)$, Kompleksitas waktu pencarian string adalah $O(n)$, dan kompleksitas waktu KMP secara keseluruhan adalah $O(m+n)$. Hal ini jauh lebih cepat dibandingkan menggunakan algoritma brute force.

Salah satu kelebihan algoritma KMP adalah ini tidak pernah berpindah ke belakang pada input text nya. Hal ini membuat algoritma ini bagus untuk memproses file besar yang dibaca dari perangkat luar. Walaupun begitu kinerja algoritma ini kurang baik saat variasi alphabet semakin banyak karena banyaknya kemungkinan mismatch dan mismatch kemungkinan terjadi di awal yang membuat algoritma menjadi kurang efektif.

Selain KMP ada juga Boyer-Moore Algorithm yang sering digunakan untuk menyelesaikan persoalan pattern matching. Algoritma ini didasarkan pada dua hal utama. Yang pertama adalah looking glass technique yaitu mencari pattern pada text dengan bergerak mundur menelusuri pattern dimulai dari

akhir pattern. Yang kedua adalah character-jump technique yaitu saat terjadi mismatch pada $T[i] \neq x$, karakter pada pattern $P[j]$ tidak sama dengan $T[i]$.

Boyer-Moore Example (1)



Skenario terburuk dari algoritma Boyer-Moore memiliki kompleksitas $O(nm + A)$. Boyer-Moore efektif jika variasi alphabetnya besar dan kurang efektif jika variasi alphabetnya sedikit. Jika dibandingkan dengan brute force, algoritma ini jauh lebih cepat saat melakukan pattern matching pada text berbahasa inggris.

Regular expression merupakan salah satu cara menyelesaikan persoalan pattern matching. Regular expression berisi sekumpulan ekspresi/karakter khusus untuk mendefinisikan pattern yang ingin dicari pada sebuah text. Regular expression dapat digambarkan dengan Finite State Machine.

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
\$	Ends with	"world\$"
*	Zero or more occurrences	"aix*"
+	One or more occurrences	"aix+"
{}	Exactly the specified number of occurrences	"al{2}"
	Either or	"falls stays"
()	Capture and group	

Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\Bain" r"ain\B"
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"
\D	Returns a match where the string DOES NOT contain digits	"\D"
\s	Returns a match where the string contains a white space character	"\s"
\S	Returns a match where the string DOES NOT contain a white space character	"\S"
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"

Set	Description
[arn]	Returns a match where one of the specified characters (a , r , or n) are present
[a-n]	Returns a match for any lower case character, alphabetically between a and n
[^arn]	Returns a match for any character EXCEPT a , r , and n
[0123]	Returns a match where any of the specified digits (0 , 1 , 2 , or 3) are present
[0-9]	Returns a match for any digit between 0 and 9
[0-5][0-9]	Returns a match for any two-digit numbers from 00 and 59
[a-zA-Z]	Returns a match for any character alphabetically between a and z , lower case OR upper case
[+]	In sets, + , * , . , , () , \$, {} has no special meaning, so [+] means: return a match for any + character in the string

BAB III

ANALISIS PEMECAHAN MASALAH

Pada tugas besar kali ini akan dibuat sebuah aplikasi chatbot berbasis web yang dapat menerima input dari user dan melakukan suatu aksi berdasarkan input yang diterima. Terdapat 8 fitur utama yang disediakan oleh chatbot. Untuk menggunakan setiap fitur, user harus memasukkan salah satu dari daftar kata penting dari setiap fitur. Untuk mengidentifikasi fitur yang diminta oleh user, aplikasi akan menggunakan pattern matching dengan algoritma KMP untuk mencocokkan input dari user dengan salah satu daftar kata penting. Setelah fitur yang ingin digunakan diketahui, aplikasi akan melakukan ekstraksi argumen - argumen yang diperlukan (seperti tanggal, kode mata kuliah, dan jenis tugas) untuk menggunakan fitur tersebut. Proses ekstraksi dilakukan dengan memanfaatkan regular expression. Untuk data - data task yang digunakan akan disimpan ke database yang berbasis sql.

Fitur pertama adalah menambahkan task baru. Untuk bisa menggunakan fitur ini, dalam input text harus terdapat salah satu dari kata berikut : “ingatkan” , “ingatin” , “ingat”. Selain itu user juga perlu menyertakan jenis dari tugas, nama mata kuliah, deskripsi dan tanggal deadline dari tugas tersebut. Jika hal - hal yang diperlukan telah dipenuhi, aplikasi akan menambahkan task tersebut pada database untuk bisa diakses dengan fitur - fitur lainnya.

Fitur kedua adalah melihat daftar task yang harus dikerjakan. Fitur ini terbagi menjadi 4 bagian yaitu menampilkan seluruh task yang terdaftar, menampilkan task pada rentang dua tanggal berbeda, menampilkan task untuk n hari/ n minggu ke depan, dan menampilkan task yang deadlinenya hari ini. Untuk menampilkan seluruh task, dalam input user harus terdapat kata “sejauh ini”. Jika dalam input diberikan jenis task yang spesifik maka aplikasi hanya akan menampilkan seluruh task yang sesuai dengan jenis yang diinput. Jika jenis task tidak diberikan, maka aplikasi akan menampilkan seluruh task yang ada pada database. Untuk menampilkan task pada rentang dua tanggal berbeda, dalam input user harus terdapat salah satu dari kata berikut : “antara”, “dari”, “hingga”, “sampai”. Selain itu, input user juga harus mengandung 2 buah tanggal berbeda yang dijadikan sebagai rentang waktu. Untuk menampilkan task untuk n hari/ n minggu ke depan, user harus menyertakan input berupa sebuah bilangan N dan sebuah kata kunci “hari” atau “minggu”. Jika diberikan jenis task yang ingin dilihat pada input, aplikasi hanya akan menampilkan jenis task tersebut. Untuk fitur menampilkan task yang deadlinenya hari ini, user dalam input yang dimasukkan harus ada kata “hari ini”.

Fitur ketiga adalah melihat deadline dari suatu tugas tertentu. Untuk bisa menggunakan fitur ini, input user harus mengandung kata “kapan”. Setelah itu aplikasi akan melakukan ekstraksi jenis task dan nama mata kuliah yang ingin dilihat deadlinenya. Jika semua berhasil, aplikasi akan menampilkan tanggal deadline dari task yang ingin dilihat.

Fitur keempat adalah memperbarui task tertentu. Kata kunci untuk fitur ini adalah “diundur”. Aplikasi akan mengekstraksi jenis task dan nama mata kuliah dari task, serta tanggal yang baru untuk mengganti tanggal yang sudah ada. jika nama mata kuliah dan jenis task terdaftar aplikasi akan memperbarui tanggal dari task tersebut pada database.

Fitur kelima adalah menandai bahwa suatu task sudah selesai dikerjakan. Fitur ini akan bisa digunakan jika dalam input terdapat kata kunci “selesai” atau “hapus”. Fitur ini juga membutuhkan

nama task dan jenis task yang ingin dihapus. Kedua informasi tersebut akan diekstraksi menggunakan regular expression. Jika semua informasi yang diperlukan terdapat pada input user, program akan menghapus tugas yang sudah selesai dari database.

Fitur keenam dan ketujuh adalah menampilkan help, dan daftar kata penting. Untuk fitur help user dapat melihat daftar fitur yang tersedia. Sementara untuk fitur daftar kata penting, user dapat melihat daftar kata penting yang perlu dimasukkan saat input agar dapat menggunakan fitur - fitur yang ada.

Fitur kedelapan adalah menampilkan pesan error. Fitur ini akan muncul jika user memasukkan input yang tidak dapat dikenali oleh aplikasi/ bot.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

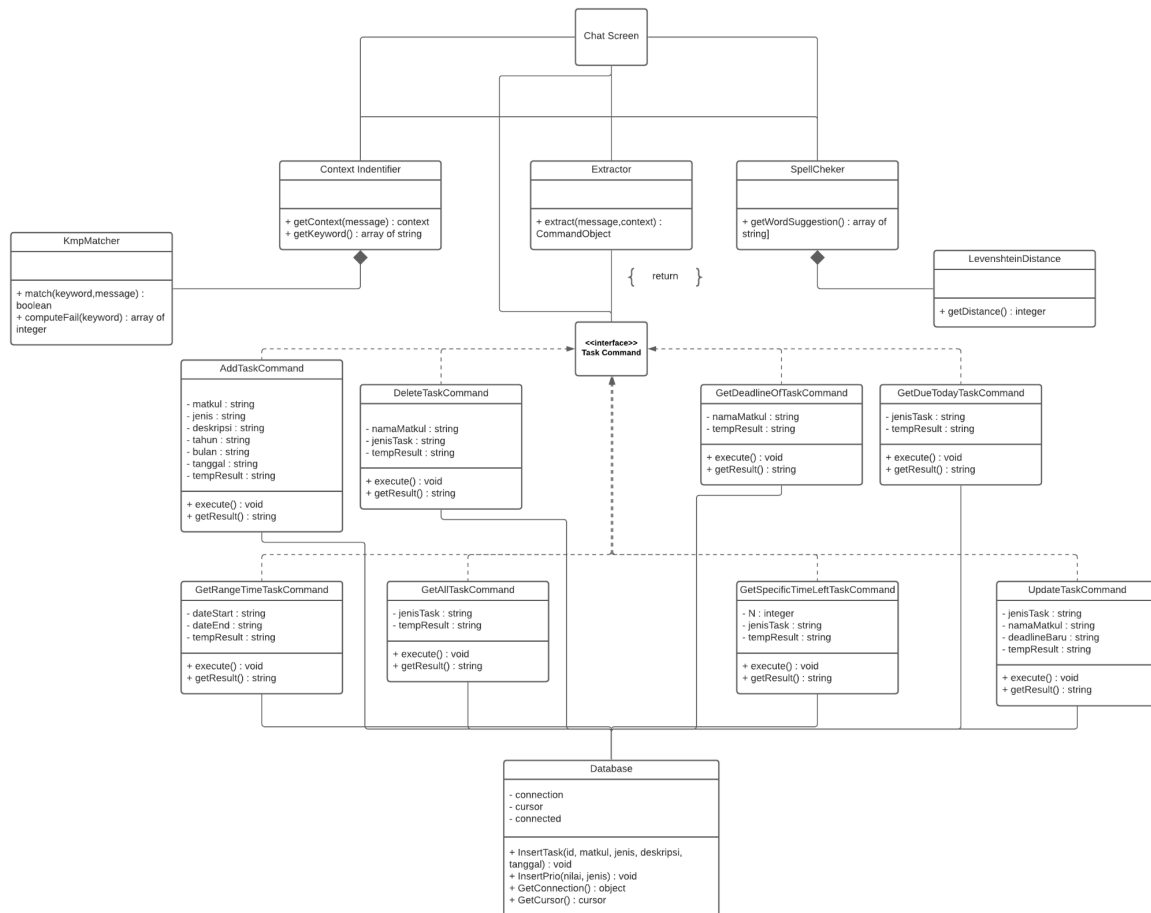
Implementasi

Pada tugas besar kali ini digunakan flask untuk frontend dan python untuk backend. Data - data yang digunakan disimpan dengan sqlite3. Untuk pengujian kode program digunakan pytest. Terdapat 8 kelas command berbeda yang dibagi berdasarkan fitur - fitur yang diperlukan aplikasi. Terdapat sebuah kelas database yang digunakan untuk melakukan proses - proses yang berhubungan dengan database. Entry point untuk flask app yang dibuat terdapat pada file app.py. Untuk melakukan pattern matching digunakan algoritma KMP yang disimpan dalam sebuah class KmpMatcher. Untuk perhitungan levenshtein distance disimpan dalam suatu class LevenshteinDistance. Setiap request yang akan dilakukan oleh akan diproses oleh objek dari class Extractor yang akan menentukan objek yang harus dibuat beserta method yang harus dipanggil.

Alur kerja program

- User memasukkan input text pada website
- Setelah input diterima objek dari kelas Extractor akan mengartikan text yang diinput user
- Akan dibuat sebuah objek dari salah satu kelas command
- Jika semua argumen yang diperlukan untuk suatu fitur dipenuhi, aplikasi akan mengeluarkan pesan sesuai fitur yang direquest oleh user

Berikut struktur kelas yang digunakan untuk membuat aplikasi ini :



Pengujian

1. Menambahkan task baru

Ingatkan saya untuk mengerjakan tubes IF2220 Regresi yang deadline-nya 19 April 2021

Task berhasil ditambahkan! ^o^

(ID: 1) IF2220 - tubes - Regresi - 2021-04-19

Bot, tolong tambahkan tubes IF2220 tentang Regresi. Deadline: 28/04/2021

Saya tidak paham :-.

Bot, tolong tambahkan tubes IF2220 tentang Regresi pada 28/04/2021

Task berhasil ditambahkan! ^o^

(ID: 2) IF2220 - tubes - tentang Regresi - 2021-04-28

2. a. melihat semua task yang ada

bot ada tugas apa saja sejauh ini ya ?

[Daftar Deadline]

(ID: 5) IF2231 - pr - string matching - 27/04/2021
(ID: 2) IF2230 - kuis - string matching - 28/04/2021
(ID: 9) IF2240 - kuis - deskripsi 4 - 2021-05-01
(ID: 3) IF2240 - ujian - string matching - 29/04/2021
(ID: 4) IF2210 - tucil - string matching - 30/04/2021
(ID: 7) IF2220 - ujian - deskripsi 2 - 2021-04-29
(ID: 8) IF2230 - tucil - deskripsi 3 - 2021-04-30
(ID: 11) IF2230 - ujian - UAS - 2021-05-20
(ID: 1) IF2221 - tubes - string matching - 27/04/2021
(ID: 10) IF2250 - tubes - deskripsi 5 - 2021-05-02

b. Pada periode tertentu (DATE_1 until DATE_2)

Apa saja deadline antara 03/04/2021 sampai 15/05/2021

[Daftar Deadline]

(ID: 2) IF2230 - kuis - string matching - 2021-04-28
(ID: 9) IF2240 - kuis - deskripsi 4 - 2021-05-01
(ID: 3) IF2240 - ujian - string matching - 2021-04-29
(ID: 4) IF2210 - tucil - string matching - 2021-04-30
(ID: 7) IF2220 - ujian - deskripsi 2 - 2021-04-29
(ID: 8) IF2230 - tucil - deskripsi 3 - 2021-04-27
(ID: 1) IF2221 - tubes - string matching - 2021-04-27
(ID: 10) IF2250 - tubes - deskripsi 5 - 2021-05-02

c. melihat task N hari ke depan

ada tugas apa saja 2 hari ke depan

[Daftar Deadline]

(ID: 7) IF2220 - ujian - deskripsi 2 - 2021-04-29
(ID: 8) IF2230 - tucil - deskripsi 3 - 2021-04-30

d. melihat task N minggu ke depan

ada tugas apa saja 2 minggu ke depan

[Daftar Deadline]

(ID: 9) IF2240 - kuis - deskripsi 4 - 2021-05-01

(ID: 7) IF2220 - ujian - deskripsi 2 - 2021-04-29

(ID: 8) IF2230 - tucil - deskripsi 3 - 2021-04-30

(ID: 10) IF2250 - tubes - deskripsi 5 - 2021-05-02

e. melihat task hari ini

Deadline tubes hari ini apa saja, ya, bot?

[Daftar Deadline]

(ID: 2) IF2220 - tubes - tentang Regresi - 2021-04-28 Tetap semangat! ^ _ ^

Tucil hari ini apa saja? plz plz

Saya tidak paham .-.

Deadline tucil hari ini apa saja, ya?

Tidak ada tucil untuk hari ini, yay! :D

3. menampilkan deadline dari suatu task tertentu

Deadline tucil IF2230 itu kapan ?

2021-04-30 - deskripsi 3

4. memperbarui task tertentu

Deadline tucil IF2230 diundur menjadi 02/02/2021

Yay deadline tugas telah diperbaharui menjadi 2021-02-02 !!!

5. menandai bahwa suatu task sudah selesai dikerjakan

bot ujian IF2230 sudah selesai ya gausah disimpan lagi

Yay tugas sudah selesai!!!

6. menampilkan opsi help yang difasilitasi oleh assistant

help

Terdapat beberapa hal yang dapat dilakukan:

- Menambah tugas (coba "Tolong ingatkan kalau ada kuis IF3110 Bab 2 pada 22/04/21")
- Melihat semua tugas (coba "bot tugas apa saja sejauh ini ya?")
- Melihat tugas pada periode tertentu (coba "Apa saja deadline antara 03/04/2021 sampai 15/04/2021")
- Melihat tugas beberapa hari/minggu ke depan (coba "Ada tugas apa saja 2 hari ke depan")
- Melihat tugas yang deadline-nya hari ini (coba "Deadline tucil hari ini apa saja, ya?")
- Menampilkan deadline dari suatu tugas tertentu (coba "Deadline tucil IF2230 itu kapan?")
- Memperbarui tugas (coba "Deadline tucil IF2230 diundur menjadi 02/02/2021")
- Menghapus/menyelesaikan tugas (coba "bot ujian IF2230 sudah selesai ya jadi gausah disimpan lagi")

Kata kunci:

- kuis
- tubes
- tucil
- ujian

7. mendefinisikan list kata penting terkait apakah itu merupakan suatu task atau tidak sudah include di nomor 6
8. menampilkan pesan error jika assistant tidak dapat mengenali masukan user

bot menurutmu tugas IF2211 itu kebanyakan ga sih ?

Saya tidak paham .-

9. Chatbot dapat memberikan rekomendasi kata jika terdapat kesalahan kata

tolong ingatkdn saya ada UAS IF2230 pada 20 Mei 2021. Saya sedang chaos nih. :

Mungkin maksud kata kunci Anda: ingatkan

BAB V

KESIMPULAN DAN SARAN

Kesimpulan

Setelah mengerjakan Tugas Besar 3 ini, didapatkan bahwa algoritma KMP dapat diterapkan untuk menyelesaikan permasalahan pattern matching dan regex dapat digunakan untuk mengekstrak informasi dari inputan yang diberikan oleh user.

Saran

Semoga untuk tugas - tugas kedepannya waktu pengerjaan dapat ditambah, karena banyaknya tugas lain yang deadlinenya berdekatan dengan tugas ini. Selain itu bagi yang akan mengembangkan aplikasi ini, dapat menambahkan kamus kata agar daftar pesan yang bisa dikenali semakin banyak.

Refleksi

Penulis beranggapan bahwa pengerjaan tugas ini kurang maksimal karena hanya dikerjakan dalam waktu 2 hari. Untuk itu dalam pengerjaan tugas - tugas berikutnya dapat mengalokasikan waktu lebih banyak agar pengerjaan hasil yang dicapai menjadi maksimal.

Referensi

1. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf> (diakses pada 28 April 2021)
2. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf> (diakses pada 28 April 2021)
3. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Tugas-Besar-3-IF2211-Strategi-Algoritma-2021.pdf> (diakses pada 28 April 2021)