

**Tugas Besar III IF2211 Strategi Algoritma
Semester II Tahun 2020/2021
Penerapan String Matching dan Regular Expression dalam
Pembangunan Deadline Reminder Assistant**



Oleh :

Farhan Nur Hidayat D - 13519071
Harith Fakhiri Setiawan - 13519161
M. Ibnu Syah Hafizh - 13519171

**PROGRAM STUDI TEKNIK 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

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

Apa saja deadline yang dimiliki sejauh ini?

[Daftar Deadline]
1. (ID: 1) 14/04/2021 - IF2211 - Tubes - String Matching
2. (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

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

[Daftar Deadline]
1. (ID: 2) 14/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

Deadline 3 minggu ke depan apa saja?

[Daftar Deadline]
1. (ID: 1) 14/04/2021 - IF2211 - Tubes - String Matching
2. (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

Deadline 1 hari ke depan apa saja?

Tidak ada

3 minggu ke depan ada kuis apa saja?

[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

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

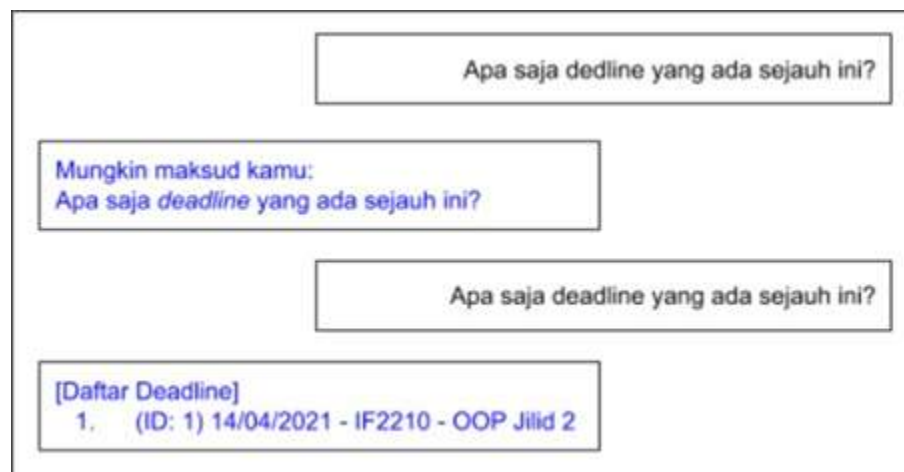
Deadline tugas IF2211 itu kapan?

14/04/2021

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 metrik 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.

Spesifikasi Program:

1. Aplikasi yang dibuat berbasis web (wajib) dan anda dapat menggunakan salah satu kaskas website: PHP, Flask, Django, JavaScript.
2. Aplikasi (backend) harus menggunakan algoritma pencocokan string KMP, Boyer-Moore, dan Regex dengan menggunakan bahasa yang menunjang regular expression: Java, Javascript, PHP, Python.
3. Penyimpanan data-data dan pengetahuan yang diperlukan oleh Chatbot bisa didefinisikan melalui 2 cara (pilih salah satu), yaitu:

- a. Membuat suatu database sederhana (penerapan Basis Data dalam Strategi Algoritma). Implementasi skema database (relasi, atribut) dibebaskan. Skema basis data tidak perlu dinormalisasi.
 - b. Menyimpannya dalam bentuk struktur data sendiri, pengambilan data dilakukan dengan menggunakan mekanisme load / save dari suatu file .txt. Struktur penyimpanan data dibebaskan.
4. Data-data yang diperlukan dan akan disimpan dalam suatu chatbot adalah sebagai
 - a. List kata-kata penting
 - b. Daftar task yang tercatat oleh Chatbot
 - c. Data-data pendukung lainnya (kreativitas kelompok)
5. Pencocokan string dapat anda implementasikan sesuai kriteria berikut.
 - a. Deteksi perintah (contoh: “Apa saja deadline yang ada sejauh ini?”) tidak dilakukan secara exact matching (input dibebaskan ke user --bukan programmer-- selama mengandung kata kunci tertentu), anda dapat memanfaatkan regular expression dan string matching untuk mencari kata kunci dan melakukan pencocokan.
 - b. Rekomendasi kata: pencocokan exact matching (KMP, Boyer-Moore) dimanfaatkan untuk menentukan tingkat kemiripan suatu kata di perintah. Anda dapat mengembangkan algoritma yang telah diajarkan untuk menentukan kemiripan string.
 - c. Pengekstrakan nilai-nilai berjenis numerik dan tanggal dilakukan dengan memanfaatkan Regular Expression.

BAB II

LANDASAN TEORI

2.1. String Matching

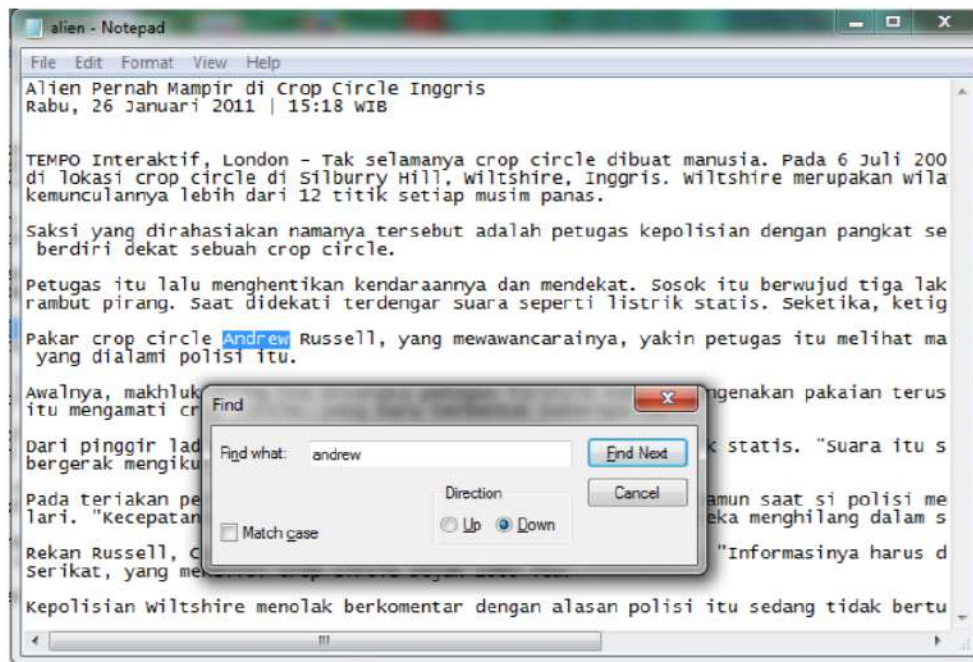
Secara mendasar, String Matching merupakan proses pencocokan sebuah string, yang memerlukan teks dan juga pattern. Teks merupakan string panjang yang memiliki n buah karakter dan merupakan string yang dicocokkan berdasarkan pattern tertentu. Lalu, teks merupakan string yang memiliki m buah karakter dengan asumsi $m \ll n$ dan merupakan string yang dicari kecocokannya ada sebuah teks.

Contoh :

1. Teks : Aku adalah seorang kapiten hehe
Pattern : kapiten
2. Teks : dia adalah seorang wibu tulen yang pemalu
Pattern : wibu

_____ Aplikasi dari String matching antara lain :

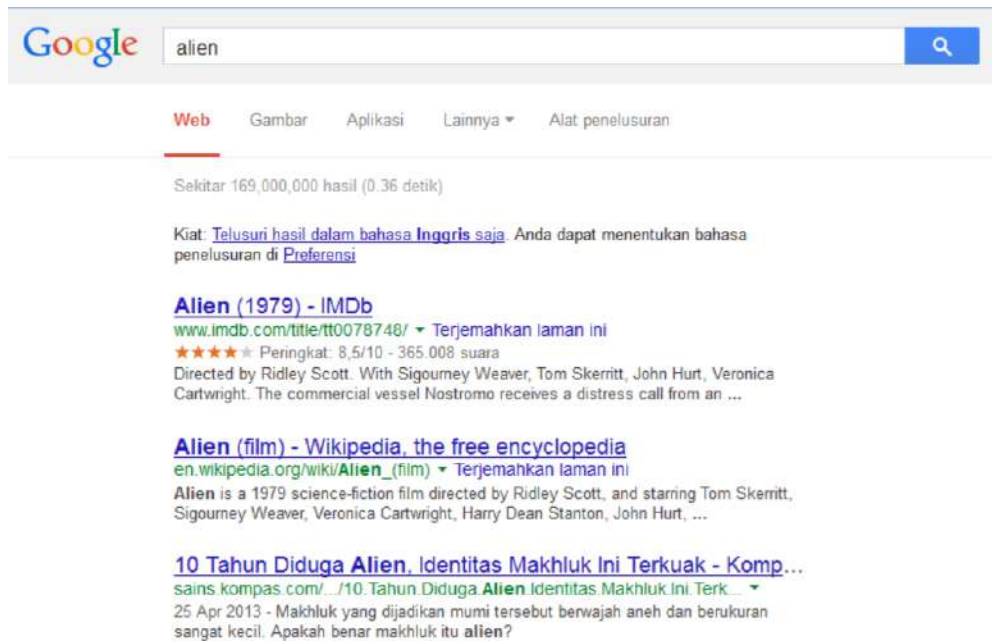
- Pencarian di dalam text editor



Gambar 2.1.1. Pencarian pada text editor

Sumber : [Slide kuliah IF2211 Strategi Algoritma : String Matching](#)

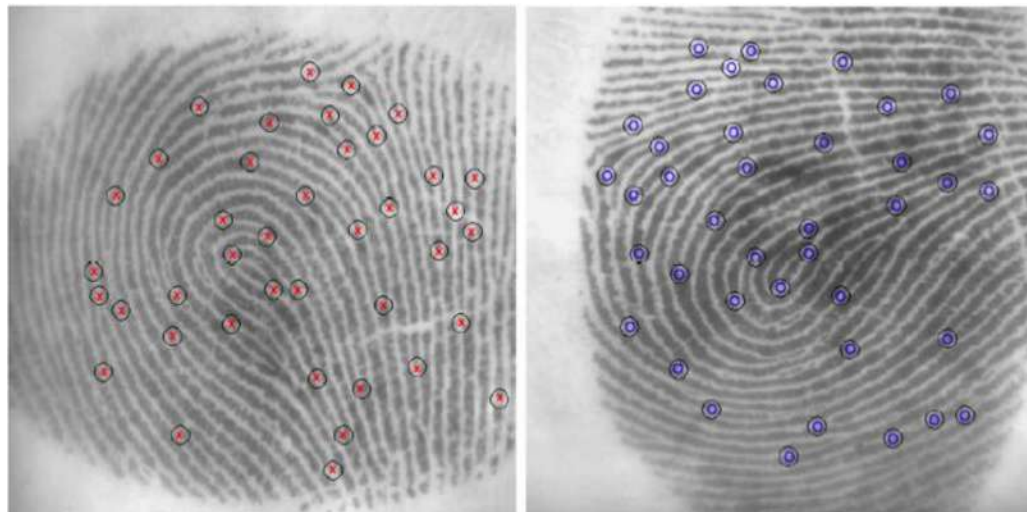
- Web Search Engine



Gambar 2.1.2. Search Engine

Sumber : [Slide kuliah IF2211 Strategi Algoritma : String Matching](#)

- Analisis citra, dan lain lain.



Gambar 2.1.3. Analisis Citra

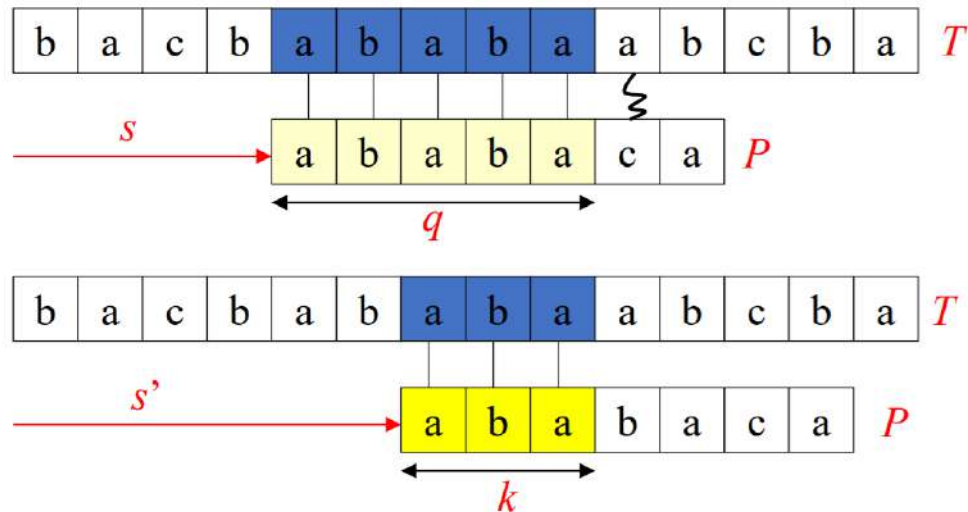
Sumber : [Slide kuliah IF2211 Strategi Algoritma : String Matching](#)

1. Algoritma Knuth-Morris-Pratt (KMP)

Algoritma Knuth-Morris-Pratt merupakan salah satu algoritma yang digunakan dalam pencarian string atau string matching. Pada algoritma Knuth-Morris-Pratt, pencarian pattern pada teks dicari dengan aturan dari kiri ke kanan seperti algoritma brute force. Namun, Algoritma Knuth-Morris -Pratt melakukan pergeseran atau penelusuran secara lebih cerdas.

Berbeda dengan algoritma brute force yang melakukan pergeseran ke kiri sebanyak satu indeks pada teks sebagai start indeks pada pattern, algoritma KMP melakukan pergeseran sebanyak mungkin sehingga langkah pengerjaannya lebih efisien dibandingkan algoritma brute force. Pergeseran pada algoritma KMP dicari menggunakan pencocokan antara prefix terbesar yang juga merupakan suffix.

Ilustrasi pengerjaan algoritma Knuth-Morris-Pratt :



Gambar 2.2.1. Analisis Citra

Sumber : [Slide kuliah IF2211 Strategi Algoritma : String Matching](#)

Pencocokan string dimulai dari 'aba' karena 'aba' merupakan prefix terpanjang yang juga merupakan suffix.

Kompleksitas Algoritma Knuth-Morris-Pratt adalah $O(m+n)$.

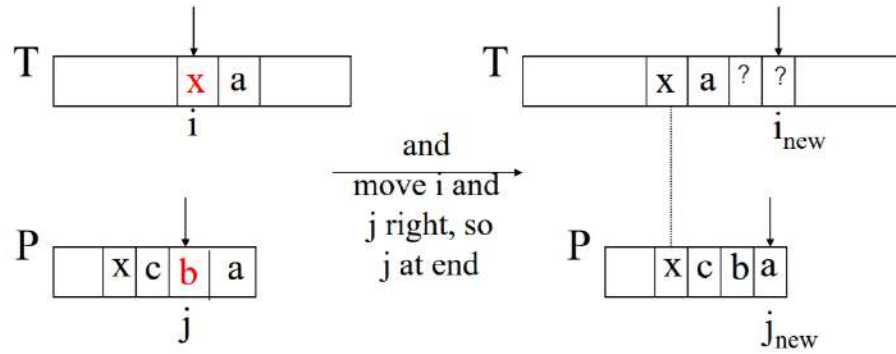
2. Algoritma Boyer Moore (BM)

Algoritma Boyer Moore merupakan salah satu algoritma yang digunakan dalam pencarian string atau string matching. Berbeda dari algoritma brute force dan knuth-morris-pratt, algoritma boyer moore menggunakan 2 teknik dalam penelusuran string pattern pada teks, yaitu *looking-glass technique* dan *character jump technique*.

looking-glass technique adalah teknik yang memeriksa kecocokan string pattern pada teks dengan dimulai dari indeks terakhir pattern. Sementara *character jump technique*, merupakan teknik yang diaplikasikan ketika terjadi mismatch antara teks[i] dan pattern[j]. Pada teknik ini terdapat 3 kasus yang mungkin terjadi. Antara lain:

Misal x adalah posisi terjadinya mismatch character, indeks i merupakan indeks terjadinya mismatch character pada teks, dan indeks j merupakan indeks terjadinya mismatch character pada pattern.

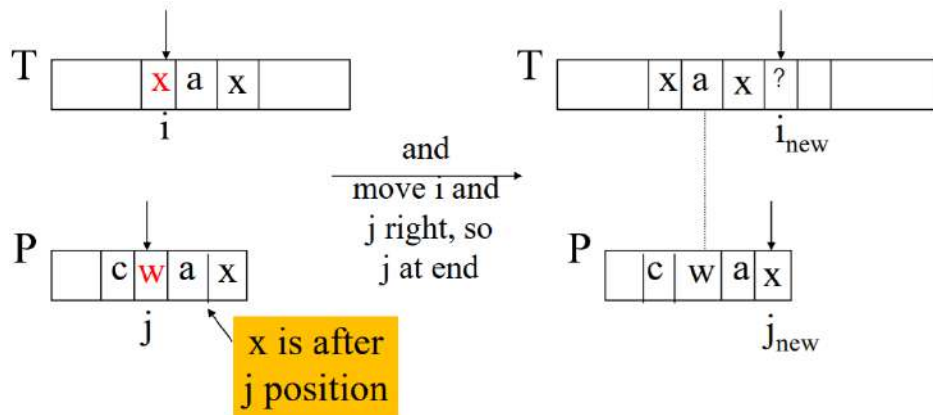
- Kasus 1 : Apabila pattern mengandung x sebelum indeks j, maka lakukan pergeseran ke kanan untuk mencocokkan x pada pattern dengan Teks[i].



Gambar 2.2.1. Ilustrasi kasus 1

Sumber : [Slide kuliah IF2211 Strategi Algoritma : String Matching](#)

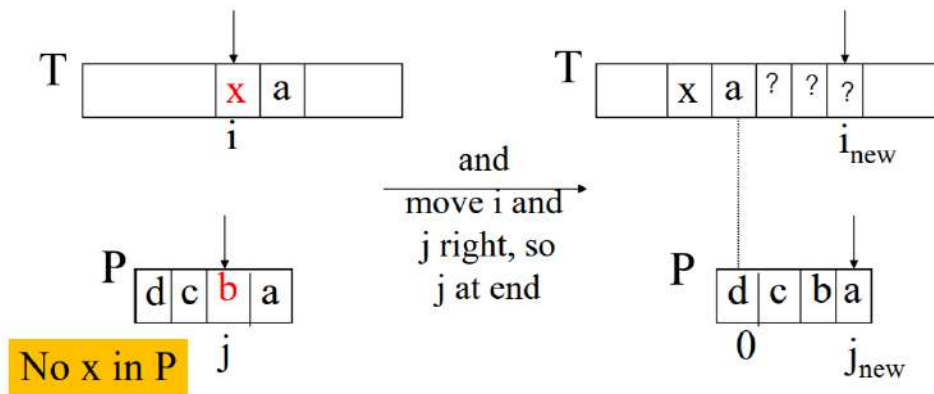
- Kasus 2 : Apabila pattern mengandung x setelah indeks j , sehingga tidak mungkin untuk melakukan pergeseran ke kanan untuk pencocokan x pada pattern dengan $\text{Teks}[i]$. Maka geser pattern ke kanan 1 karakter menuju $\text{Teks}[i+1]$



Gambar 2.2.2. Ilustrasi kasus 2

Sumber : [Slide kuliah IF2211 Strategi Algoritma : String Matching](#)

- Kasus 3 : Selain kasus 1 dan kasus 2, geser pattern untuk mencocokkan $\text{pattern}[0]$ dengan $\text{Teks}[i+1]$.



3. Regular Expression

Regular expression atau regex merupakan sebuah urutan karakter yang mengekspresikan atau menspesifikasikan sebuah pola pencarian. Umumnya, pola regular expression ini digunakan untuk pencarian, dan juga “find and replace” yang dioperasikan pada string. Selain itu, regular expression juga dapat digunakan untuk memvalidasi sebuah masukan atau input.

Character classes pada regex :

Construct	Deskripsi
[abc]	a, b, atau c (simple class)
[^abc]	Semua karakter selain a,b,c (negasi)
[a-zA-Z]	a sampai z atau A sampai Z, inclusive (range)
[a-d[m-p]]	a sampai d atau m sampai p (gabungan)
[a-z&&[def]]	d, e atau f (irisan)
[a-z&&[^bc]]	a sampai z, kecuali b dan c (subtraksi)
[a-z&&[^m-p]]	a sampai z, dan bukan m sampai p (subtraksi)

Predefined Character Class pada regex :

Construct	Deskripsi
.	Semua karakter
\d	Digit [0-9]
\D	Non digit [^0-9] (hati-hati dengan huruf besar)
\s	Whitespace character [\t\n\r\x0B\f]
\S	Non whitespace character [^s]
\w	Word character [a-zA-Z_0-9]
\W	Non word character [^\w]

Quantifier pada regex :

Construct	Arti
-----------	------

X?	X muncul satu atau tidak sama sekali
X*	X muncul nol atau banyak
X+	X muncul satu atau banyak
x{n}	X muncul tepat n kali
x{n,}	X muncul setidaknya n kali
x{n,m}	X muncul antara n sampai m kali

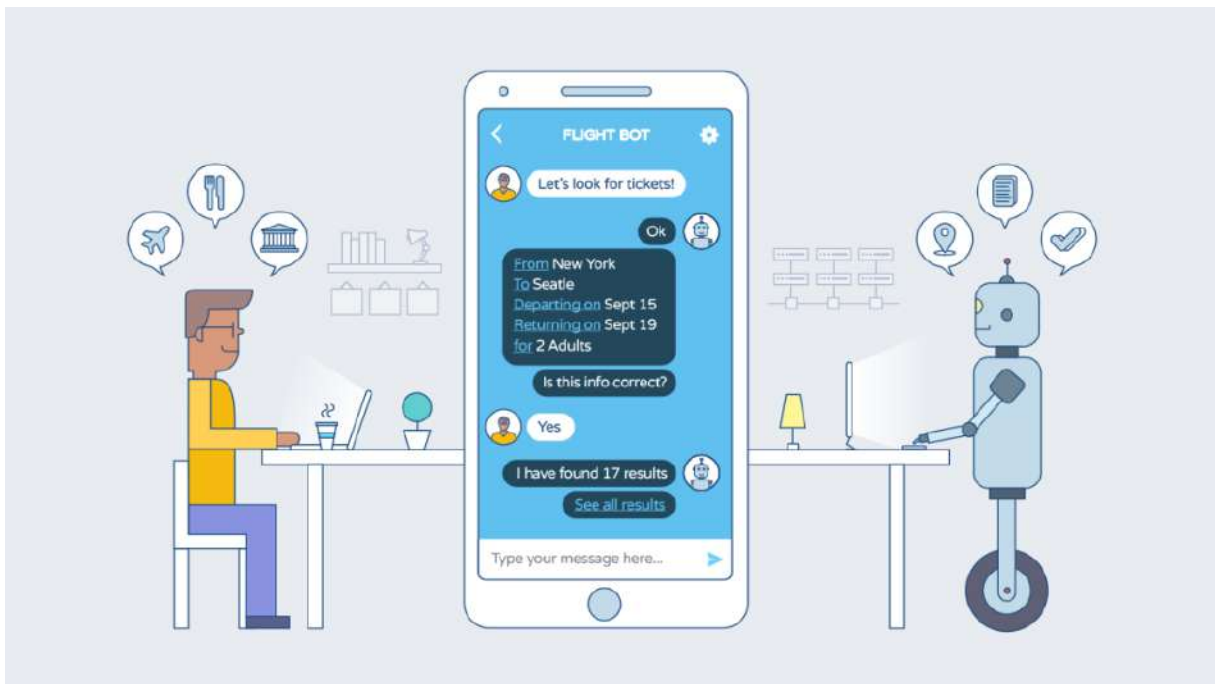
Boundary Matchers pada regex :

Construct	Deskripsi
^	Awal baris
\$	Akhir baris
\b	Batas kata
\B	Batas bukan kata
\G	Akhir match sebelumnya
\Z	Akhir dari input tapi untuk final terminator jika ada
\z	Akhir dari input

4. Chatbot

Chatbot merupakan sebuah bot yang telah terintegrasi sesuai tujuannya. Sesuai dari nama chatbot sendiri, antarmuka dengan pengguna dilakukan melalui *chatting* atau berbasis chat. Chatbot memiliki berbagai fungsi sesuai dengan tujuannya tersendiri. Beberapa chatbot antara lain :

- Chatbot pada proses jual beli online
- Chatbot pada layanan customer service
- Chatbot untuk schedule remainder, dan lain lain.



Gambar 2.4.1. Chatbot

Sumber : groeducontentmarketing.com

Kelebihan dari chatbot :

- Respon cepat, tidak terpengaruh kesibukan dari bot kecuali terjadi *server down*.
- Mempermudah layanan 24/7.

BAB III

ANALISIS PEMECAHAN MASALAH

3.1 Langkah penyelesaian masalah setiap fitur

3.1.1. Fungsi regex

1. cleanQuery : fungsi ini digunakan untuk menghapus spasi berlebih dan mengubah query menjadi lowercase.
2. cariTanggal : fungsi ini digunakan untuk mencari tanggal pada query, format tanggal yang benar adalah DD-MM-YYYY, fungsi akan mengembalikan array berisi tanggal-tanggal yang terdapat pada query.
3. cariAngkaSelainTanggal : fungsi ini digunakan untuk mencari angka-angka pada query yang bukan merupakan tanggal, sehingga angka dengan format tanggal tidak akan dibaca.
4. cariKodeMatkul : fungsi ini digunakan untuk mencari kode mata kuliah pada query, suatu kata didefinisikan sebagai kode mata kuliah apabila terdiri dari dua huruf dan empat angka, contoh : IF2211, KU2931.

3.1.2. Fungsi String matching

1. createPrefix : fungsi ini digunakan untuk membuat list of prefix dari string masukkan.
2. createSubfix : fungsi ini digunakan untuk membuat list of subfix dari string masukkan.
3. takeFirstN : fungsi ini digunakan untuk membuat pengambilan n karakter utama dari sebuah string masukkan.
4. getGeser : fungsi ini dibuat untuk menentukan pergeseran yang dilakukan ketika terjadi mismatch antara pattern[j] dan Teks[i] berdasarkan list of prefix dan list of subfix sesuai Knuth-Morris-Pratt.
5. KMPsearch : Fungsi ini dibuat untuk menentukan apakah sebuah string Teks mengandung string pattern dengan memanfaatkan algoritma Knuth-Morris-Pratt. Mengembalikan true apabila ditemukan kecocokan string dan false apabila tidak ditemukan.

3.1.3. Fungsi Fitur

1. Menambahkan task baru

Suatu query atau masukan pengguna akan diklasifikasikan sebagai perintah untuk menambahkan hal baru apabila masukan mengandung empat komponen wajib yaitu tanggal, kode mata kuliah, jenis tugas, dan topik tugas. Fitur ini diimplementasikan pada fungsi rAddTask dan fungsi AddTask.

Setelah query dimasukkan, fungsi rAddTask akan dijalankan dengan parameter query tersebut. Pertama akan dilakukan pembersihan query menggunakan fungsi cleanQuery.. Kedua, akan dilakukan pencarian tanggal menggunakan regex dengan fungsi cariTanggal, Jika array tidak mengandung tepat satu tanggal, dalam arti lain query tidak mengandung tanggal atau mengandung lebih dari satu tanggal, maka fungsi akan berhenti dan fitur untuk menambahkan task tidak akan dijalankan. Ketiga, akan dilakukan KMP pada query untuk mencari apakah query mengandung kata penting yang sudah didefinisikan pada array kataPenting. Jika tidak ditemukan kata penting, maka program akan berhenti.

Keempat, akan dilakukan regex untuk mencari kode mata kuliah pada query. Kelima, akan dicari topik dari task, dimana topik dari task pasti ditulis setelah kode mata kuliah. Hanya jika keempat komponen ditemukan akan dilakukan pemasukkan data ke database menggunakan fungsi AddTask. Apabila pengguna sudah pernah memasukkan data yang sama persis, maka akan diberikan pesan bahwa task sudah pernah dicatat pada nomor id sekian.

2. Melihat daftar task yang harus dikerjakan

Untuk melihat daftar task, query harus mengandung kata “deadline”, tetapi ada kasus khusus dimana tidak perlu kata “deadline” untuk melihat daftar task, tetapi query wajib mengandung semua komponen berikut : kata Penting, tanggal, dan kata “minggu” atau “hari”. Contoh query pada kasus khusus adalah “3 minggu lagi ada kuis apa saja ?” atau “2 hari kedepan ada tubes apa ?”. Fungsi yang dipakai adalah seeTask dan rSeeTask.

Pengecekan apakah query termasuk kasus khusus akan dilakukan diawal, jika bukan kasus khusus maka akan dilakukan KMP untuk mencari kata “deadline”. Kata penting jenis task tidak wajib ada. Setelahnya akan dilakukan pencarian tanggal menggunakan regex, apabila ada dua tanggal pada query, maka akan ditampilkan daftar deadline dari periode tanggal pertama sampai tanggal kedua dan fungsi berhenti.

Jika tanggal tidak berjumlah dua, maka akan dilakukan regex untuk mencari angka dengan fungsi cariAngkaSelainTanggal, dimana ini dilakukan untuk mencari jumlah hari atau jumlah minggu kedepan yang mau dilihat deadlinenya. Setelahnya akan dilakukan KMP untuk mencari kata “minggu”, jika ditemukan akan ditampilkan deadline sesuai jumlah minggu kedepan, dan fungsi berhenti. Jika tidak ditemukan, akan dicari kata “hari” untuk menampilkan deadline jumlah hari kedepan dan program berhenti. Jika kata “hari” atau “minggu” tidak ditemukan dan query tidak mengandung tanggal, akan digunakan KMP untuk mencari kata “hari ini” untuk menampilkan deadline hari ini atau kata “besok” untuk menampilkan deadline besok. Apabila masih tidak ditemukan, maka akan ditampilkan semua deadline yang ada.

3. Menampilkan deadline dari suatu task tertentu

Untuk menampilkan deadline suatu task tertentu, task wajib berjenis tugas, yaitu tucil atau tubes. Fungsi yang dipakai adalah seeDeadline dan rSeeDeadline.

Pertama akan dicari kata “deadline” menggunakan KMP. Kedua, masih menggunakan KMP, akan dicari kata “tugas”, “tubes”, atau “tucil”. Ketiga, akan dicari kode mata kuliah menggunakan regex, dimana hanya boleh ada satu kode mata kuliah pada query. Apabila semua komponen terpenuhi, maka akan dicari data yang sesuai pada database dan ditampilkan pada chatbot.

4. Memperbaharui task tertentu

Memperbaharui task tertentu menggunakan fungsi updateTask dan rUpdateTask. Untuk menjalankan fitur, query wajib mengandung kata “deadline”, satu buah tanggal, kata “task”, dan id task tersebut. Pencarian kata “deadline” dan “task” menggunakan fungsi KMP, pencarian tanggal menggunakan regex pada fungsi cariTanggal, dan pencarian id task menggunakan regex pada fungsi cariAngkaSelainTanggal. Jika semua komponen

terpenuhi, maka task dengan id tersebut pada database tanggalnya akan diperbaharui dengan tanggal pada query. Setelahnya akan ditampilkan pesan bahwa update berhasil.

5. Menandai bahwa suatu task sudah selesai dikerjakan

Menandai suatu task tertentu menggunakan fungsi markTask dan rMarkTask. Untuk menjalankan fitur, query wajib mengandung salah satu dari kata sinyal yaitu “udah”, “kerja”, “mengerja”, atau “selesai”. Query juga harus mengandung kata “task” yang dicari menggunakan KMP, dan id task yang dicari menggunakan regex pada fungsi carAngkaSelainTanggal. Jika semua komponen terpenuhi, maka pada database, kolom ‘Status’ pada task tersebut nilainya akan diubah menjadi 1, artinya task sudah dikerjakan. Setelahnya akan disampaikan pesan bahwa mark task tersebut berhasil.

6. Menampilkan opsi help

Apabila query mengandung kata atau frasa “help”, “apa saja”, “apain”, atau “lakukan”, dan tidak mengandung tanggal, kata penting, dan kode matkul, maka query akan didefinisikan sebagai query untuk menampilkan pesan help. Kata-kata sinyal tersebut dicari menggunakan KMP. Jika ditemukan maka akan chatbot akan menampilkan hal-hal yang bisa dilakukan oleh bot.

7. Mendefinisikan list kata penting

Kata penting yang digunakan untuk penentuan jenis tugas dari suatu task dibuat static pada array bernama kataPenting, dengan elemen “Kuis”, “Ujian”, “Tucil”, “Tubes”, dan “Praktikum”.

8. Menampilkan pesan error

Program utama memiliki sebuah boolean isRun yang diinisialisasi dengan False. Ketika sebuah fungsi pada fitur-fitur diatas dijalankan, apabila fungsi berjalan sampai selesai, maka boolean isRun akan diubah menjadi True. Jadi ketika program dijalankan dan tidak ada fungsi yang selesai, maka isRun akan tetap bernilai False dan pesan error akan ditampilkan.

3.2. Fitur fungsional dan arsitektur Chatbot yang dibangun

1. Textfield input

Textfield untuk input pada bagian bawah digunakan untuk menuliskan query input.



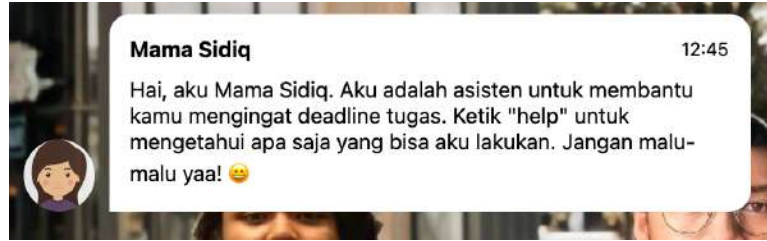
2. Send button

Send button adalah button yang diklik ketika pengguna ingin mengirimkan query input. Selain send button, pengguna juga dapat mengirim query input menggunakan tombol “enter” pada keyboard.



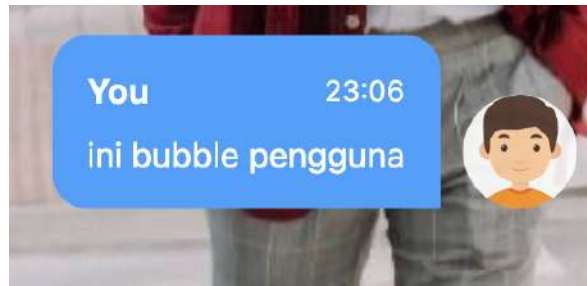
3. Bubble mamasidiq

Bubble mamasidiq menampilkan pesan atau informasi yang dibalas oleh mamasidiq. Bubble ini berwarna putih dengan avatar perempuan.



4. Bubble pengguna

Bubble pengguna menampilkan pesan atau query yang dikirim oleh pengguna kepada mamasidiq bot. Bubble berwarna biru dengan avatar seorang laki-laki.



BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Struktur data

Struktur data yang digunakan yang digunakan pada chat bot assistant ini adalah string dan array of string. String merepresentasikan teks masukan dari user dan array of string dimanfaatkan untuk menyimpan kata-kata penting. Terdapat berbagai fungsi dan prosedur yang digunakan untuk membangun chat bot assistant. Fungsi dan prosedur tersebut dan kegunaannya adalah:

1. cleanQuery
Parameter : String
Kegunaan : Menghapus spasi berlebih dan mengubah teks menjadi lower case
2. cariTanggal
Parameter : String
Kegunaan : Mencari tanggal pada teks dengan regular expression
3. cari AngkaSelainTanggal
Parameter : String
Kegunaan : Mencari angka selain tanggal pada teks dengan regular expression
4. cariKodeMatkul
Parameter : String
Kegunaan : Mencari kode matkul pada teks dengan regular expression
5. createPrefix
Parameter : String
Kegunaan : Mengembalikan list prefix dari string masukkan

6. createSubfix
Parameter : String
Kegunaan : Mengembalikan list subfix dari string masukan
7. takeFirstN :
Parameter : String
Kegunaan : Mengembalikan string dari karakter pertama sampai karakter ke-n
8. getGeser
Parameter : array, array
Kegunaan : Mengembalikan besar shifting yang dilakukan ke kanan sesuai dengan input list of subfix dan list of prefix.
9. KMPSearch
Parameter : String, string
Kegunaan : Mencari kata (string 1) pada teks (string 2) dengan menggunakan metode Knuth Morris Pratt dan mengembalikan true jika terdapat kata pada teks atau mengembalikan false jika tidak terdapat kata pada teks.
10. rAddTask
Parameter : String, boolean (apakah terdapat suatu fungsi yang sedang berjalan atau tidak)
Kegunaan : Menambah task baru
11. rSeeTask
Parameter : String
Kegunaan : Melihat daftar task dengan batas waktu ataupun tidak
12. rShowDeadline
Parameter : String
Kegunaan : Melihat daftar task yang hanya bersifat tugas (tubes, tucil)
13. rUpdateTask
Parameter : String
Kegunaan : Memperbarui waktu task (diundur/diubah/diganti)
14. rMarkTask
Parameter : String
Kegunaan : Menandakan suatu task sudah dikerjakan. Task dihapus dari database.
15. rHelp
Parameter : String
Kegunaan : Menampilkan fitur-fitur yang dapat dilakukan oleh assistant bot.
16. main
Parameter : String
Kegunaan : Menjalankan program utama.

Fungsi-fungsi atau prosedur diatas bukanlah semua fungsi atau prosedur pada program, melainkan fungsi atau prosedur penting yang merupakan inti dari program. Terdapat fungsi-fungsi atau prosedur lainnya untuk membantu fungsi atau prosedur diatas, namun hanya berperan sebagai fungsi atau prosedur pembantu fungsi atau prosedur penting dan tidak berpengaruh langsung pada program utama.

4.2. Tata cara penggunaan program

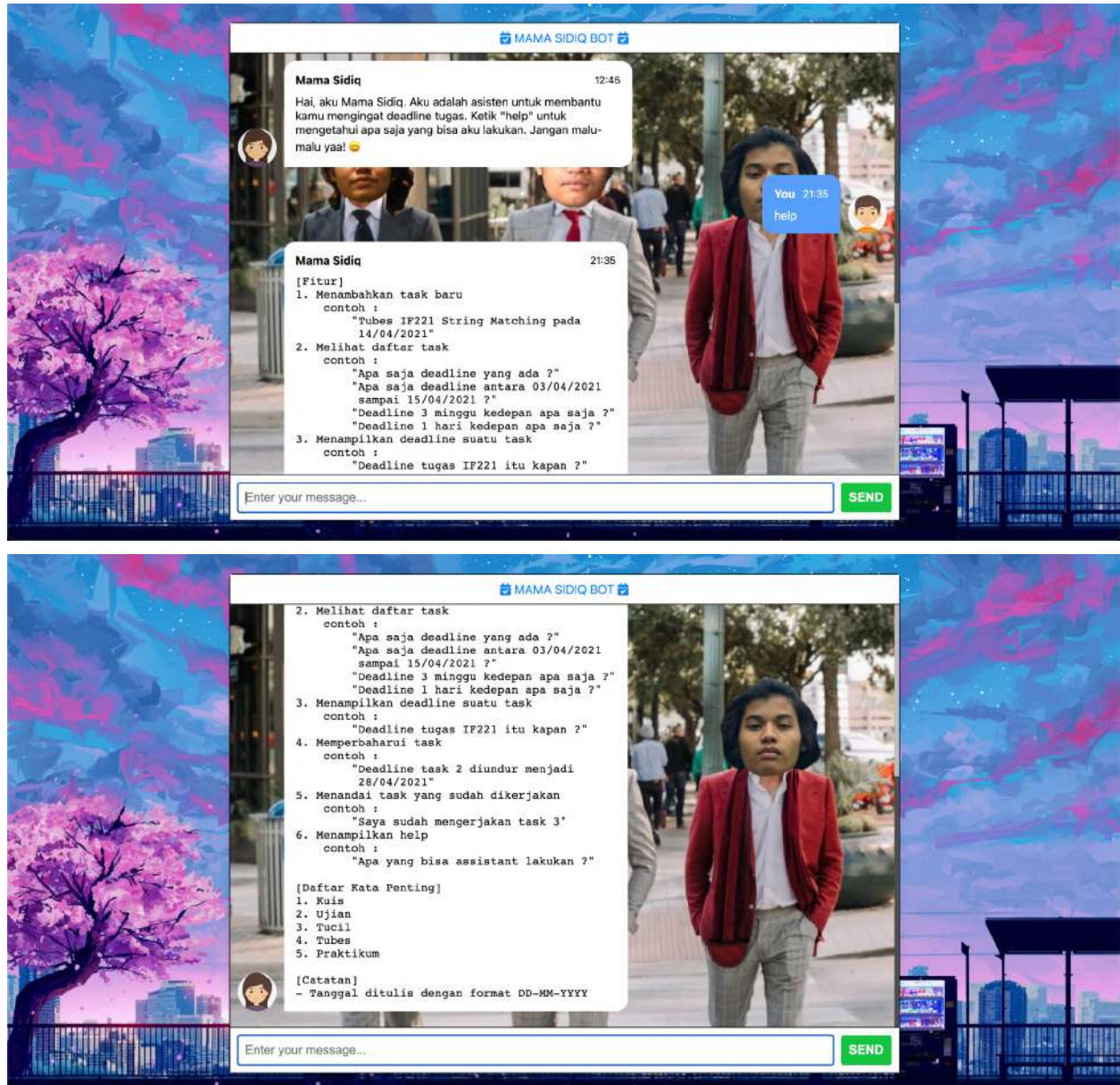
Sebelum menjalankan chatbot ini, terdapat beberapa requirements untuk menunjang keberhasilan berjalannya program. Requirements tersebut antara lain sudah menginstall Python, sudah menginstall flask. Langkah-langkah untuk menjalankan program chatbot adalah sebagai berikut:

- a. Mac OS:
 1. Buka terminal,
 2. Pindah ke directory src,
 3. Ketik “Python3 app.py”,
 4. Buka browser, lalu pergi ke "localhost:5000".
- b. Windows
 1. Buka cmd,
 2. Pindah ke directory src,
 3. Ketik “python app.py”,
 4. Buka browser, lalu pergi ke "localhost:5000".

Kemudian pada ” localhost:5000” akan ditampilkan halaman sebagai berikut,



Ketik “help” untuk melihat fitur-fitur yang tersedia,

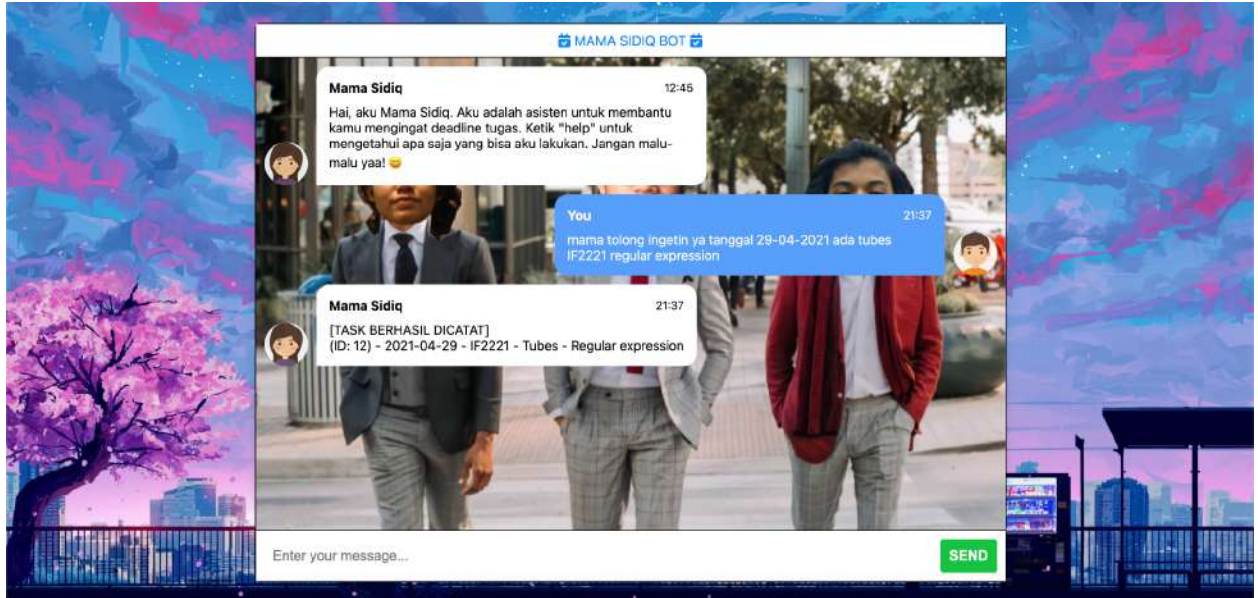


Selanjutnya fitur-fitur bot dapat digunakan dengan mengirimkan pesan seperti contoh yang tertulis pada help.

4.3. Hasil pengujian dan analisis

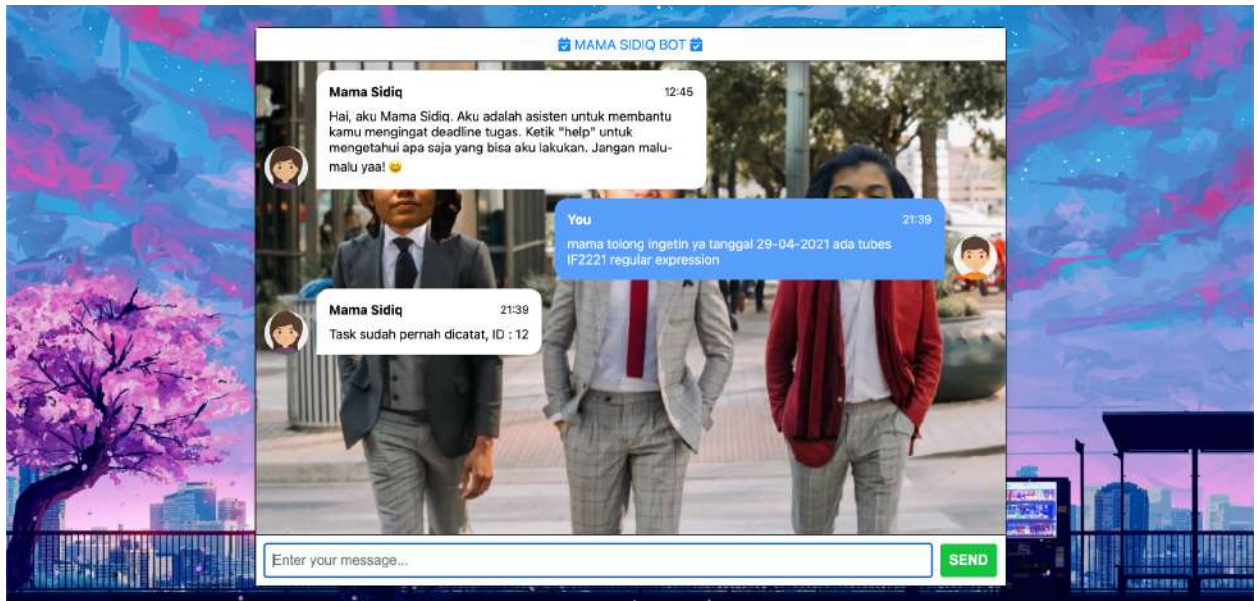
4.3.1. Menambahkan task baru

- Berikut adalah tangkapan layar fitur menambah task baru jika task belum terdaftar:



Program berhasil menemukan tanggal, salah satu dari kata penting, kode mata kuliah, dan topiknya pada query. Program juga tidak menemukan data yang sama persis pada database, sehingga data ditambahkan ke database dan diberikan keluaran seperti diatas.

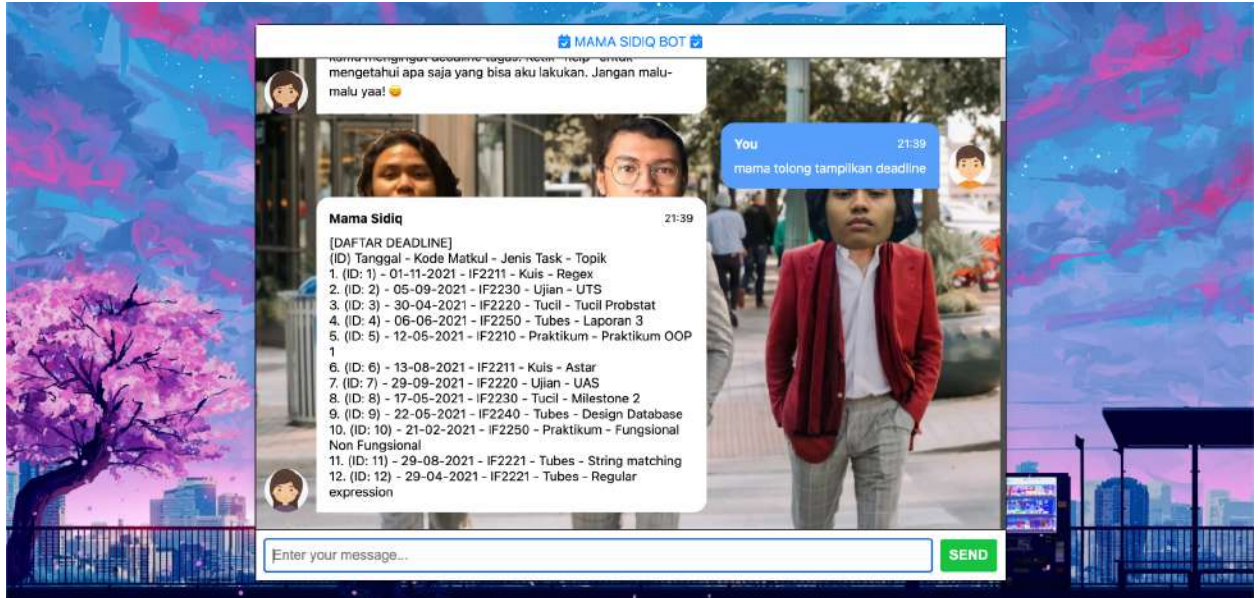
- Berikut adalah tangkapan layar fitur menambah task baru jika task sudah terdaftar:



Program berhasil menemukan tanggal, salah satu dari kata penting, kode mata kuliah, dan topiknya pada query. Program menemukan data yang sama di database pada ID 12, sehingga data tidak dimasukkan ke database dan diberi keluaran seperti diatas.

4.3.2. Melihat daftar task yang harus dikerjakan

Berikut adalah tangkapan layar fitur melihat daftar task:



Program hanya menemukan kata kunci “deadline”, sehingga menampilkan seluruh deadline yang ada.

4.3.3. Menampilkan deadline dari suatu task tertentu

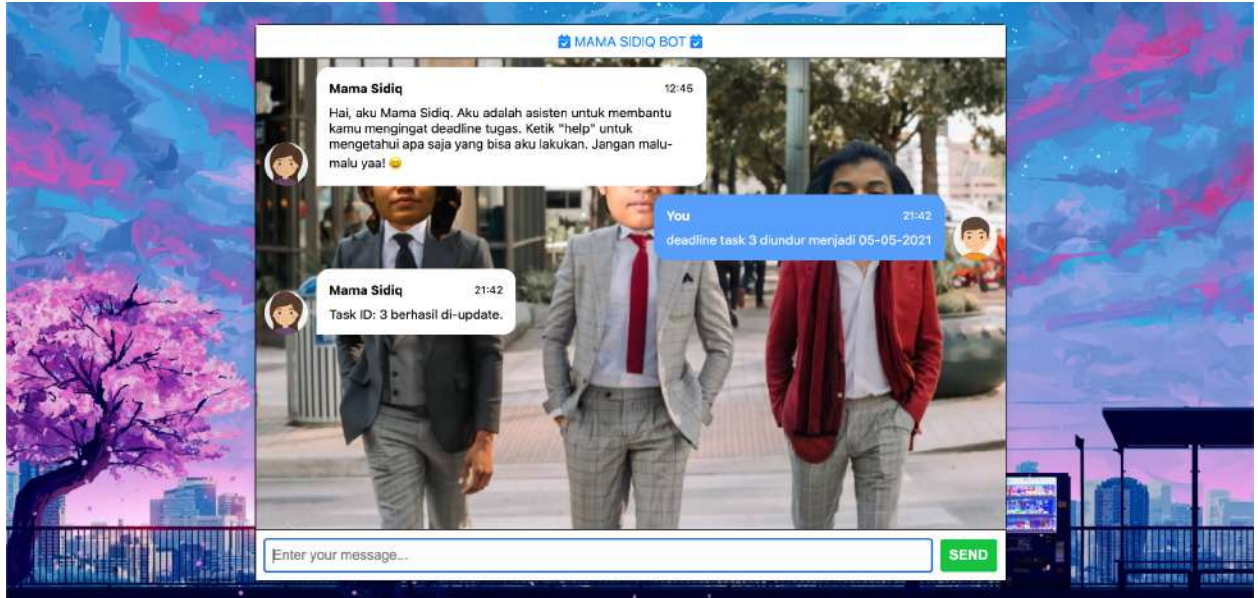
Berikut adalah tangkapan layar fitur menampilkan deadline dari suatu task tertentu:



Program menemukan kata “deadline”, kata penting yaitu “tugas”, dan kode mata kuliah, sehingga menjalankan fungsi rShowDeadline dan menampilkan deadline tugas (tubes dan/atau tugil) yang ada pada mata kuliah tersebut.

4.3.4. Memperbarui task tertentu

Berikut adalah tangkapan layar fitur memperbarui task tertentu:



Program menemukan kata “deadline”, kata “task”, nomor dari task, dan tanggal, sehingga menjalankan fungsi `rUpdateTask`. Karena task 3 sudah tercatat di database, maka diberikan keluaran pesan bahwa update berhasil.

4.3.5. Menandai bahwa suatu task sudah selesai dikerjakan

Berikut adalah tangkapan layar fitur menandai suatu task sudah selesai dikerjakan:



Program menemukan kata “task”, nomor dari task, dan salah satu dari kata sinyal pada fungsi `rMarkTask`, sehingga fungsi tersebut dijalankan. Kata sinyal pada fungsi ini antara lain 'udah', 'kerja', 'mengerja', dan 'selesai'. Karena task 8 sudah tercatat di database, maka diberikan keluaran pesan bahwa task berhasil ditandai.

4.3.6. Menampilkan fungsi help yang difasilitasi oleh assistant

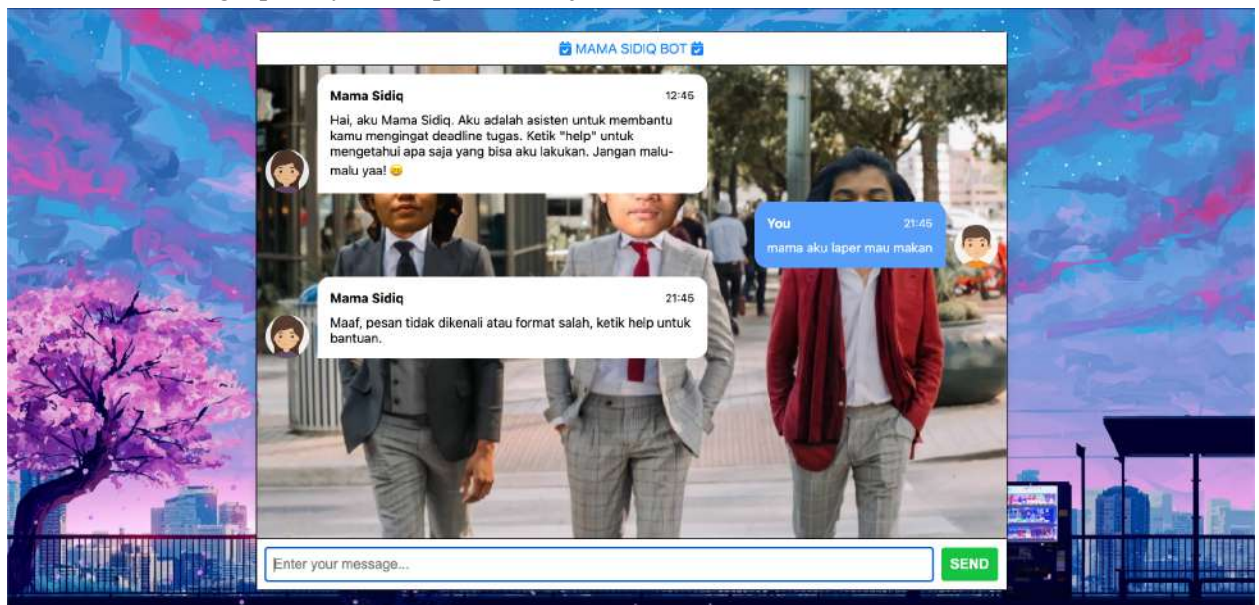
Berikut adalah tangkapan layar fitur help:



Program menemukan salah satu dari kata sinyal fungsi rHelp yaitu 'help', 'apa saja', 'apain', dan 'lakukan'. Program tidak menemukan kata “deadline”, kata penting, kode mata kuliah, dan tanggal, sehingga fungsi rHelp dijalankan dan diberi keluaran sesuai diatas.

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

Berikut adalah tangkapan layar fitur pesan error jika masukan user tidak dikenali:



Program tidak menemukan kata kunci yang memenuhi fungsi-fungsi yang ada, sehingga tidak ada fungsi yang dijalankan, dan diberi keluaran seperti diatas.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. Algoritma Knuth-Morris-Pratt dapat digunakan untuk mencari suatu pattern tertentu pada suatu teks. Proses pencarian pattern ini dilakukan dengan memanfaatkan algoritma yang mirip dengan algoritma brute force, namun lebih efisien dengan kompleksitas waktu $O(m+n)$. Pencocokan string pada KMP ini dilakukan dengan melakukan *shift* atau pergeseran ke kanan sebanyak mungkin untuk membuat proses lebih efisien. Proses pergeseran ini ditentukan oleh prefix terpanjang yang juga merupakan bagian dari suffix ketika terjadi mismatch.
2. Regex dapat digunakan untuk mencari sebuah pola tertentu dari suatu teks. Pada tugas kali ini, regex digunakan untuk menentukan teks yang berupa tanggal, kode mata kuliah, dan juga id task. Selain hanya mencari, regex juga bisa digunakan untuk “find and replace”. Pada tugas kali ini, digunakan untuk mencari input query yang berisi huruf kapital kemudian membuatnya menjadi huruf kecil, serta menghilangkan apabila terjadi double space atau lebih pada input query dan menggantinya dengan single space.

5.2. Saran

Berdasarkan proses pengerjaan dan percobaan yang telah dilakukan, penulis program berharap agar ilmu yang didapat selama pembuatan program yang telah dibuat tidak hanya menjadi pemenuhan kewajiban Tugas Besar III IF2211 Strategi Algoritma, namun terus dikembangkan. Selain itu, penulis program juga berharap konsep pemahaman terkait String Matching dan Regular Expression, ataupun hal lain yang dipelajari selama pengerjaan tugas ini dapat diaplikasikan terhadap hal lain kedepannya. Oleh karena itu, sebaiknya tugas-tugas seperti ini terus dipertahankan dan terus dibuat lebih menarik sehingga mahasiswa lebih semangat dalam mempelajari dan memahami berbagai bahasa, serta algoritma pemrograman dengan baik.

5.3. Komentar dan Refleksi

Setelah mengerjakan tugas besar ini, penulis program menjadi lebih paham mengenai pembuatan String Matching dan Regular Expression beserta pengaplikasiannya.

DAFTAR PUSTAKA

“Pencocokan String (String/Pattern Matching)”. Informatika.stei.itb.ac.id. Diakses pada 25 April 2021.

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>

“Pencocokan String dengan regular expression”. Informatika.stei.itb.ac.id. Diakses pada 25 April 2021.

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>

Lampiran

Video :

https://drive.google.com/drive/folders/1ZM0U4HgSM_oBTMZTHBZRUX-NgBF9N6HC?usp=sharing

Video dibuka menggunakan akun std.