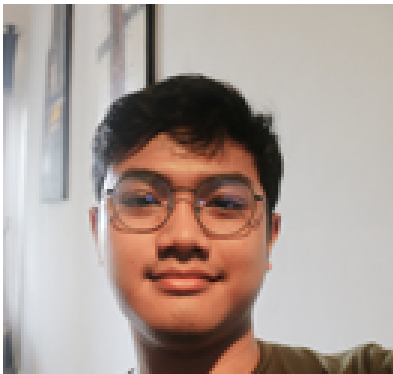


TUGAS BESAR IF2211 STRATEGI ALGORITMA

Penerapan String Matching dan Regular Expression dalam

Pembangunan Deadline Reminder Assistant



Oleh Kelompok RickRolled

13519166 Karina Imani

13519202 Farhan Yusuf Akbar

13519220 Leo Cardhio Sih Pratama

TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

BAB 1

DESKRIPSI TUGAS

1.1 Spesifikasi 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 tugas-tugas yang ada.

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.

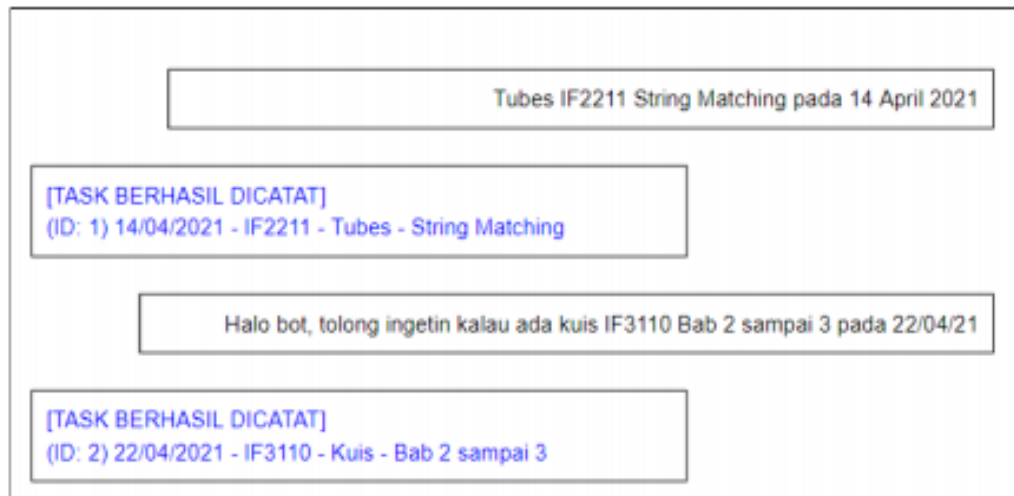
1. Menambahkan task baru

- 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

i. 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.

d. Contoh interaksi

Apa saja deadline yang dimiliki sejauh ini?

[Daftar Deadline]

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

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

[Daftar Deadline]

- (ID: 2) 14/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

Deadline 3 minggu ke depan apa saja?

[Daftar Deadline]

- (ID: 1) 14/04/2021 - IF2211 - Tubes - String Matching
- (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]

- (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

3. Menampilkan deadline dari suatu task tertentu

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

4. Memperbaharui task tertentu

- Memperbarui tanggal dari suatu task (dalam kehidupan nyata, tentu ada kejadian dimana deadline dari suatu task diundur)
- 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. Tupil
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", "Tupil", "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



BAB 2

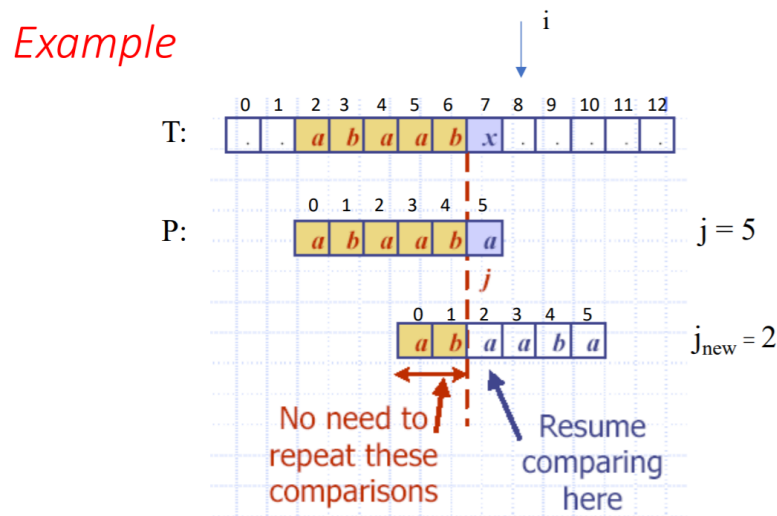
LANDASAN TEORI

2.1 Algoritma String/Pattern Matching (KMP, BM, Regex)

2.2.1 Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt atau lebih sering disingkat KMP adalah salah satu metode melakukan string/pattern matching. Algoritma ini berperilaku mirip dengan algoritma paling sederhana atau brute force dalam hal memindai teks dan pattern dari kiri ke kanan (awal ke akhir), namun dinilai lebih efektif karena ketika menemukan ketidakcocokkan (mismatch) dapat menggeser lebih dari 1 space.

Sebagai contoh, ketika pattern $P[j]$ mengalami mismatch pada teks $T[j]$, pattern dapat “digeser” sejauh prefiks $P[0..j-1]$ terbesar yang juga merupakan sufiks dari $P[1..j-1]$.



Gambar 2.1. Contoh pergeseran KMP.

Oleh karena itu, salah satu elemen algoritma ini adalah Border Function ($b[k]$), yaitu sebuah fungsi yang terlebih dahulu memproses pattern untuk menemukan kecocokan antara prefiks-prefiks P dengan P sendiri. Sebelum dilakukan string matching, akan dibuat tabel yang

berisi j sebagai posisi indeks pada P , $P[j]$ sebagai alfabet P pada posisi j , k sebagai $j-1$, dan $b[k]$ sebagai hasil dari Border Function pada k .

j	0	1	2	3	4	5
$P[j]$	a	b	a	a	b	a
k	-	0	1	2	3	4
$b(k)$	-	0	0	1	1	2

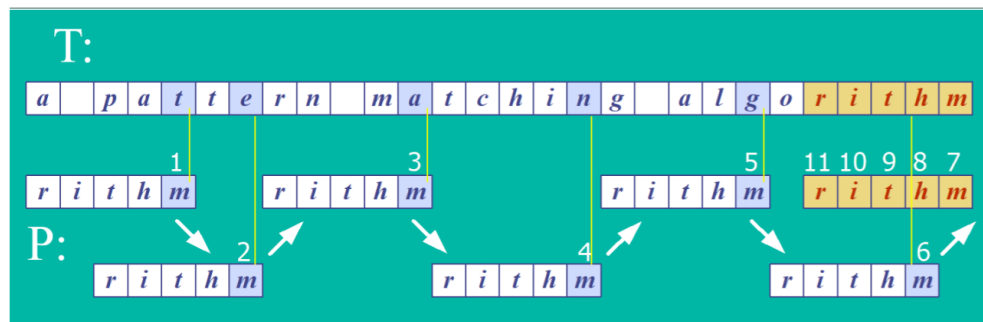
$b(k)$ is the size of the largest border.

Gambar 2.2. Contoh border function.

Kompleksitas algoritma ini adalah $O(m + n)$, dengan $O(m)$ kompleksitas border function, dan $O(n)$ kompleksitas pattern matching untuk kasus terburuk. Adapun, kelebihan KMP adalah tidak pernah bergerak mundur selama melakukan pattern matching, sehingga cocok digunakan untuk memproses file-file berukuran besar. Sedangkan, kelemahannya adalah untuk ukuran alfabet yang besar, akan seringkali terjadi mismatch sehingga kompleksitasnya menyerupai brute force (hanya menggeser 1 space).

2.2.2 Boyer-Moore

Algoritma berikutnya yang dapat digunakan untuk pattern matching adalah algoritma Boyer-Moore. Algoritma ini dapat dikatakan berbeda dari brute force dan KMP karena memindai pattern P dari kanan ke kiri (akhir ke awal), meskipun algoritma ini tetap memindai teks T dari kiri ke kanan (awal ke akhir).



Gambar 2.3. Contoh pergeseran BM.

Algoritma ini memanfaatkan dua elemen, yaitu Character Jump Technique dan Last Occurrence Function. Character jump technique menentukan seberapa banyak P akan bergeser tergantung posisi terakhir huruf tersebut dilihat pada P:

- Jika karakter mismatch terakhir ditemukan di kiri lokasi mismatch, geser sampai karakter tersebut ter-align (terluruskan) dengan karakter mismatch.
- Jika karakter mismatch terakhir ditemukan di kanan lokasi mismatch, geser seluruh pattern sebanyak 1 space (seperti brute force).
- Jika tidak ada karakter mismatch pada keseluruhan P, geser seluruh P sampai indeks 0 melewati karakter mismatch.

x	a	b	c	d
$L(x)$	4	5	3	-1

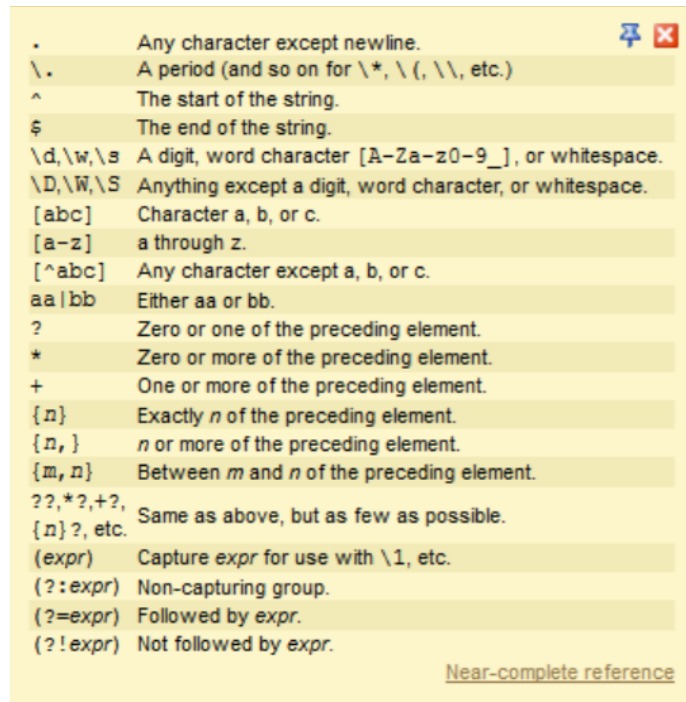
Gambar 2.4. Contoh last occurrence function.

Untuk melakukan character jump technique, dibutuhkan last occurrence function yang melakukan preprocessing pattern dan membuat array kemunculan tiap huruf T pada P. Tabel yang dihasilkan berisi x atau semua karakter unik pada P, dan $L(x)$ yaitu fungsi yang menghasilkan kemunculan terakhir x pada P. Jika x tidak terdapat pada P, $L(x)$ akan menghasilkan -1.

2.2.3 Regular Expression

Regular Expression, Regex, atau Regexp adalah salah satu cara yang paling mudah dan efektif untuk mencari dan memanipulasi string, terutama dalam memproses file-file teks berukuran besar. Regex dapat mempersingkat kode-kode panjang untuk melakukan string matching dengan menggunakan berbagai sintaks yang didukung oleh banyak bahasa pemrograman seperti Perl, Python, PHP, JavaScript, maupun Java, bahkan sampai text editor canggih seperti Microsoft Word.

Dalam pencarian string/pattern menggunakan Regex, dapat digunakan konvensi penulisan pattern khusus yang membuat pencarian dapat dilakukan dalam skala yang lebih luas dan fleksibel. Adapun, beberapa konvensi tersebut adalah:



.	Any character except newline.
\.	A period (and so on for *, \ (, \\, etc.)
^	The start of the string.
\$	The end of the string.
\d,\w,\s	A digit, word character [A-Za-z0-9_], or whitespace.
\D,\W,\S	Anything except a digit, word character, or whitespace.
[abc]	Character a, b, or c.
[a-z]	a through z.
[^abc]	Any character except a, b, or c.
aa bb	Either aa or bb.
?	Zero or one of the preceding element.
*	Zero or more of the preceding element.
+	One or more of the preceding element.
{n}	Exactly n of the preceding element.
{n, }	n or more of the preceding element.
{m, n}	Between m and n of the preceding element.
??,*?,+?	Same as above, but as few as possible.
{n} ?, etc.	
(expr)	Capture expr for use with \1, etc.
(?:expr)	Non-capturing group.
(?=expr)	Followed by expr.
(?!expr)	Not followed by expr.

[Near-complete reference](#)

Gambar 2.5. Beberapa konvensi penulisan Regex.

2.2 Chatbot

Chatbot kelompok RickRolled berbasis Django, yang memanfaatkan Python sebagai bahasa utama program, HTML/CSS sebagai penyusun GUI, SQLite untuk menyimpan database task, kata penting, dan pesan, serta sebagian kecil JavaScript, JQuery, dan Ajax untuk menjalankan fungsionalitas frontend. Chatbot dinamakan Rick, dan dapat dijalankan dengan mengetikkan `python manage.py runserver` untuk melakukan local hosting. Link yang dihasilkan kemudian dapat dibuka pada browser.

GUI chatbot memiliki 3 halaman utama, yaitu main, about, dan erase log. Main berisi chat box yang dapat digunakan untuk memulai pembicaraan dengan chatbot, about berisi informasi singkat tentang bot, dan erase log berfungsi untuk menghapus database pesan yang telah ditukar dengan bot sehingga halaman main kembali kosong.

Adapun, bot dapat menjalankan fitur-fitur yang tertera sesuai spesifikasi, seperti menambahkan task dengan mata kuliah, deadline, dan topik tertentu, melihat seluruh task yang sudah dicatat, maupun membuat query untuk mencari task dengan deadline tertentu, mata kuliah tertentu, maupun deadline pada jangka waktu tertentu. Hal ini dilakukan dengan mengetikkan query ke chatbot, dengan daftar perintah yang disediakan oleh bot.

BAB 3

ANALISIS PEMECAHAN MASALAH

3.1 Langkah penyelesaian masalah setiap fitur

1. Menambahkan task baru

Menambahkan task baru menggunakan algoritma regex untuk mencari tanggal dari task dan kode matkul. Untuk melihat kelengkapan komponen task digunakan algoritma KMP. Kelengkapan dicek dengan kata penting. Jika komponen tidak lengkap akan dikeluarkan pesan kesalahan

2. Menampilkan semua deadline

Mengakses database dan menampilkan semua task

3. Menampilkan deadline pada periode tertentu

Mengecek tanggal dari dan sampai menggunakan regex dan mencari task dengan tanggal di antara kedua tanggal tersebut

4. Menampilkan fitur yang ada

3.2 Fitur fungsional dan arsitektur Chatbot yang dibangun

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi Teknis Program (struktur data, fungsi, prosedur yang dibangun)

1. App Frontend

a. Struktur data (models.py)

i. Message:

```
class Message(models.Model) :  
    name = models.TextField()  
    text = models.TextField()  
  
    def print(self):  
        print(self.name)  
        print(self.text)
```

b. Fungsi dan prosedur (views.py)

i. Kelas Main:

Memiliki 2 fungsi yaitu get, untuk me-render template main.html dan post, untuk memperoleh input pengguna ke chat box. Fungsi post memasukkan input pengguna dan respons chatbot ke database Message untuk ditampilkan real-time.

ii. Kelas About:

Memiliki fungsi get untuk me-render template about.html.

iii. Kelas Delete:

Memiliki fungsi get untuk me-render template delete.html sekaligus menghapus semua objek Message dalam database.

iv. Kelas Messages:

Memiliki fungsi get untuk me-render halaman .json yang digunakan untuk menampilkan semua objek Message.

2. App Task

a. Struktur data (models.py)

i. Task:

```
class Task(models.Model):
    tanggal = models.DateField(blank=False)
    kodematakuliah = models.TextField(blank=False)
    jenistugas = models.TextField(blank=False)
    topikugas = models.TextField(blank=False)
```

ii. KeyWords:

```
class KeyWord(models.Model):
    word = models.TextField(blank=False)
    tipe = models.TextField(blank=False, default=None)
```

b. Fungsi dan prosedur (utils.py, views.py)

i. File utils.py

```
def keyWord_getTipe(keyWord)
# Mendapatkan tipe dari keyword input yang dideteksi oleh
pattern matching.

def task_addTask(tgl, kode, jenis, topik)
# Menambahkan task ke database berdasarkan input tanggal, kode,
jenis, dan topik.

def task_deleteTask(task_id)
# Menghapus task dengan id tertentu.

def task_getTask(task_id, tipe="*")
# Mengembalikan task dengan id tertentu.

def task_getTaskDurasi(tgl_dari, tgl_ke, tipe="*")
# Mengembalikan task dengan jangka waktu tanggal_dari sampai ke
tanggal_ke.

def kmpString (input_text, kata_penting)
# Melakukan string matching dengan metode Knuth-Morris-Pratt
(KMP) dengan input string user.

def repeatPatternCounter (pattern)
# Fungsi untuk menghitung prefiks/sufiks terbesar untuk
kebutuhan KMP.
```

ii. File views.py

Memiliki fungsi `index` yang digunakan untuk kebutuhan testing, menampilkan hasil yang diperoleh dari testing setiap fungsi penting.

```
def index(request):
    try:
        data = keyWord_getTipe("deadline")
        date = datetime.datetime(2021, 2, 21)
        task_addTask(date, "d", "c", "0")
        date_dari = datetime.datetime(2021, 2, 20)
        date_ke = datetime.datetime(2021, 2, 23)
        data_date = task_getTaskDurasi(date_dari,
        date_ke, "tanggal")
        data_task = task_getTask(5, "tanggal")
        task_deleteTask(4)
        return HttpResponse(data_date[0])

    except OperationalError:
        return HttpResponse(Task.objects.model._meta.db_table)
```

4.2 Tata Cara Penggunaan Program

1. Interface Program

Chatbot dibuat menggunakan Django, oleh karena itu, untuk menjalankan program digunakan command `python manage.py runserver` pada direktori `manage.py` (folder `src`). Link local hosting yang diberikan dapat dibuka pada browser. Pastikan template dan CSS styling berada pada folder `venv`.

Browser akan membuka halaman utama (`main.html`), dimana pengguna dapat langsung memberikan input query untuk menjalankan fitur-fitur program. Button “About” dapat diklik untuk melihat keterangan singkat tentang chatbot, dan button “Erase Log” dapat diklik untuk menghapus log percakapan dengan bot sejauh ini.

Adapun, chat log yang dibuat pada `main.html` memanfaatkan JQuery dan Ajax untuk melakukan refresh secara terus menerus, dengan interval 1000 ms atau 1 detik.

2. Fitur-fitur Program


Chatbot dapat menerima query, melakukan string/pattern matching dengan KMP dan Regex, dan menerjemahkan query tersebut menjadi perintah menambah atau menghapus task,

mencari task dengan syarat tertentu, dan sebagainya. Program memenuhi fungsi-fungsi yang tercantum dalam spesifikasi, yakni:

- ❖ Menambahkan task baru dengan format kalimat yang mencakup tanggal, kode mata kuliah, jenis tugas, dan topik.
- ❖ Melihat daftar task yang harus dikerjakan, dapat berupa seluruh task, periode tertentu, maupun jenis task.
- ❖ Menampilkan opsi help yang berisi daftar perintah yang dapat dilaksanakan oleh bot.

4.3 Hasil Pengujian (screenshot antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur dalam Chatbot)

1. Menambahkan task baru:

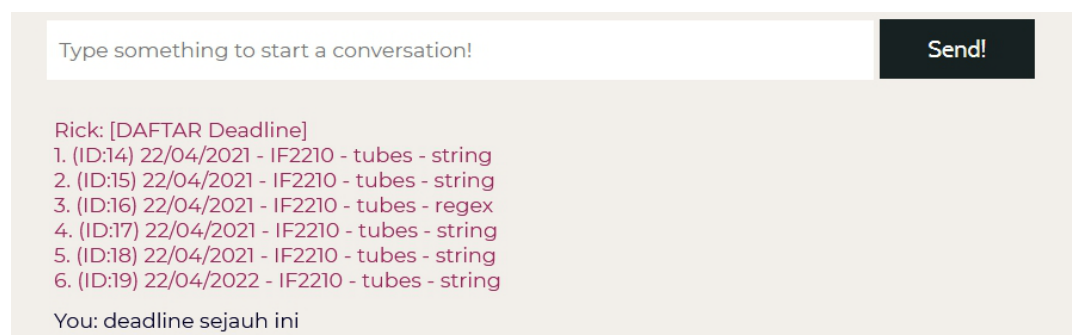


Rick: [TASK BERHASIL DICATAT]
ID: 2 - 22/04/2022 - tubes - string

You: ingatkan tubes IF2210 pada 22/04/2022 topik string

2. Melihat daftar task yang harus dikerjakan:

- a. Deadline sejauh ini:



Type something to start a conversation! Send!

Rick: [DAFTAR Deadline]

- 1. (ID:14) 22/04/2021 - IF2210 - tubes - string
- 2. (ID:15) 22/04/2021 - IF2210 - tubes - string
- 3. (ID:16) 22/04/2021 - IF2210 - tubes - regex
- 4. (ID:17) 22/04/2021 - IF2210 - tubes - string
- 5. (ID:18) 22/04/2021 - IF2210 - tubes - string
- 6. (ID:19) 22/04/2022 - IF2210 - tubes - string

You: deadline sejauh ini

- b. Deadline jangka waktu tertentu:

Type something to start a conversation!
Send!

Rick: [DAFTAR Deadline]
1. (ID:14) 22/04/2021 - IF2210 - tubes - string
2. (ID:15) 22/04/2021 - IF2210 - tubes - string
3. (ID:16) 22/04/2021 - IF2210 - tubes - regex
4. (ID:17) 22/04/2021 - IF2210 - tubes - string
5. (ID:18) 22/04/2021 - IF2210 - tubes - string
You: deadline dari 01/01/2021 sampai 05/05/2021
Rick: [DAFTAR Deadline]
1. (ID:19) 22/04/2022 - IF2210 - tubes - string
You: deadline dari 01/01/2022 sampai 05/05/2022

3. Menampilkan opsi help:

Type something to start a conversation!
Send!

Rick: [FITUR YANG TERSEDIA]
1. Menambahkan task baru
2. Melihat daftar task

Rick: [DAFTAR KATA PENTING]
1. kuis
2. uts
3. tucil
4. tubes
5. deadline
6. tanggal
7. pada
8. pr
9. topik
You: help

4.4 Analisis Hasil Pengujian

Program dapat menjalankan fitur-fitur yang didaftarkan di atas dengan baik. Adapun, fitur yang dapat dilakukan chatbot adalah menambahkan task, melihat deadline tertentu (sejauh ini dan pada jangka waktu tertentu), serta menampilkan opsi help. Sementara itu, fitur yang tidak dapat dilakukan chatbot adalah melihat deadline berdasarkan tipe task, melihat deadline task tertentu, dan menandakan task selesai (menghapus task dari database).

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

String/pattern matching menggunakan KMP dan Regex dapat dimanfaatkan untuk membuat chatbot sederhana yang dapat mengingatkan mahasiswa tentang deadline tipe-tipe task tertentu.

5.2 Saran

Saran dari kelompok penulis adalah agar semua tubes 3 yang telah dibuat oleh seluruh kelompok mahasiswa IF2211 Strategi Algoritma tahun 2021 pada tugas besar ini dapat di-publish. Tujuan dari hal tersebut adalah agar antar kelompok dapat melihat alternatif implementasi yang mungkin dapat membantu menambah wawasan dan mengetahui pola pikir yang unik.

DAFTAR PUSTAKA

- Tim Mata Kuliah IF2211 Strategi Algoritma. (2021). *Spesifikasi Tugas Besar*,
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Tugas-Besar-3-IF2211-Strategi-Algoritma-2021.pdf>, diakses 28 April 2021.
- Munir, Rinaldi. (2021). *Pencocokan String (String/Pattern Matching)*.
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses 28 April 2021.
- Munir, Rinaldi. (2021). *String Matching dengan Regular Expressions*.
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regular-Expressions-2019.pdf>, diakses 28 April 2021.
- Towards Data Science. (2020) *Pattern Search with the Knuth-Morris-Pratt (KMP) algorithm*.
<https://towardsdatascience.com/pattern-search-with-the-knuth-morris-pratt-kmp-algorithm-8562407dba5b>, diakses 28 April 2021.
- Geeks for Geeks. (2021). *Boyer-Moore Algorithm for Pattern Searching*.
<https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/>, diakses 28 April 2021.
- Chua Hock-Chuan. (2018). *Regular Expressions (Regex)*.
<https://www3.ntu.edu.sg/home/ehchua/programming/howto/Regexe.html>, diakses 28 April 2021.