

TUGAS BESAR 3 IF2211 STRATEGI ALGORITMA

Disusun dalam rangka memenuhi tugas Strategi Algoritma (IF2211)

Kelompok botkita



Disusun Oleh:

Muhammad Fadli Gunardi	13519130
Muhammad Rayhan Ravianda	13519201
Farhan Fadillah Rafi	13519204

SARJANA TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2020

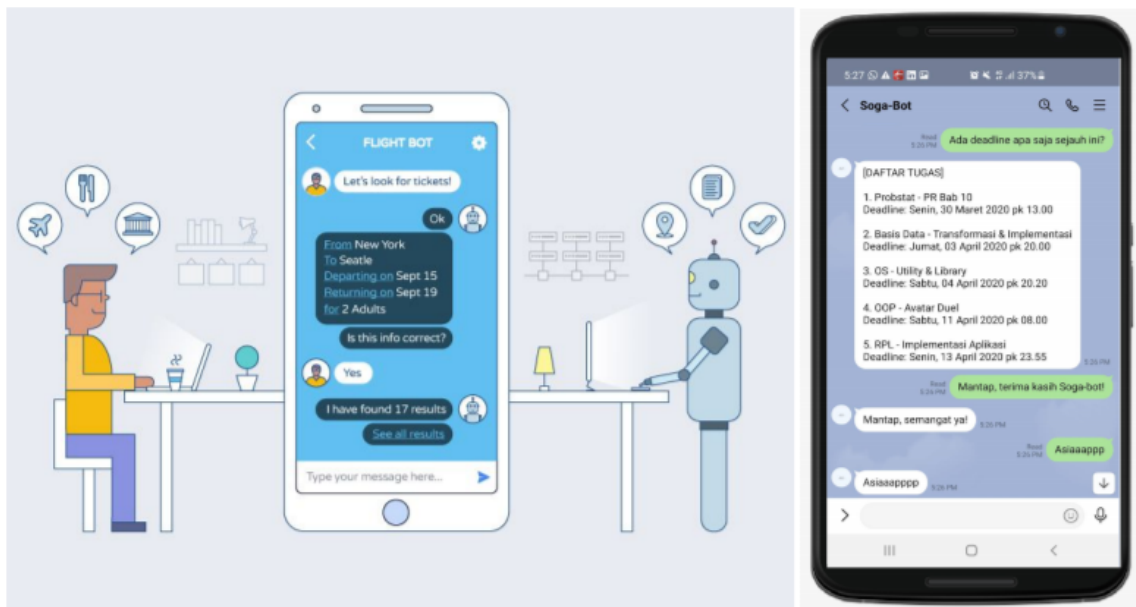
DAFTAR ISI

DAFTAR ISI	1
BAB I DESKRIPSI TUGAS	2
BAB II LANDASAN TEORI	4
2. 1 Algoritma KMP, BM, dan Regex	4
2. 2 Chatbot	7
BAB III ANALISIS PEMECAHAN MASALAH	9
3. 1 Langkah Pemecahan Masalah Setiap Fitur	9
3. 2 Fitur Fungsional dan Arsitektur Chatbot	11
BAB IV IMPLEMENTASI DAN PENGUJIAN	12
4. 1 Spesifikasi Teknis Program	12
4. 2 Penjelasan Tata Cara Penggunaan Program	18
4. 3 Hasil Pengujian	18
4. 4 Analisis Hasil Pengujian	21
BAB V KESIMPULAN DAN SARAN	22
5. 1 Kesimpulan	22
5. 2 Saran	22
5. 3 Refleksi	22
DAFTAR PUSTAKA	23

BAB I

DESKRIPSI TUGAS

Bukan sesuatu yang janggal lagi jika semakin hari tugas-tugas di Teknik Informatika Semester 4 semakin bertambah banyak. Hal ini tentunya berakibat pada bertambahnya kegiatan dan pekerjaan yang harus dilakukan mahasiswa. Tak jarang pula ada tugas yang terlupakan karena mahasiswa sulit untuk mengingat semua tugas dan deadline tersebut. Oleh karena itu, mahasiswa Teknik Informatika berniat untuk membuat suatu Google Assistant sederhana berupa Deadline Reminder Assistant, atau dalam bahasa Indonesia adalah Asisten Pengingat Deadline.



Gambar 1. Ilustrasi Chatbot dan Asisten Pengingat Deadline

Sumber : <https://id.pinterest.com/pin/824299538024636729/> dan dokumentasi pribadi

Di era digital ini, kita tentu sudah pernah mendengar teknologi atau aplikasi seperti Chatbot, LINE Bot, atau Google Assistant. Ketiganya merupakan agen cerdas yang meniru kemampuan manusia untuk melakukan percakapan dengan user. Kehadiran Chatbot ini tentu membantu kehidupan manusia, khususnya dalam membantu menyajikan informasi yang diperlukan user dan menjawab berbagai pertanyaan yang sering

ditanyakan oleh user. Secara spesifik dalam konteks Asisten Pengingat Deadline ini, Chatbot tersebut akan menjawab pertanyaan-pertanyaan mahasiswa yang sering ditanyakan seperti deadline seminggu ke depan, deadline di bulan ini, dan task-task penting lainnya yang perlu dilakukan. Chatbot ini akan sangat membantu user agar tidak lagi melewatkan deadline 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

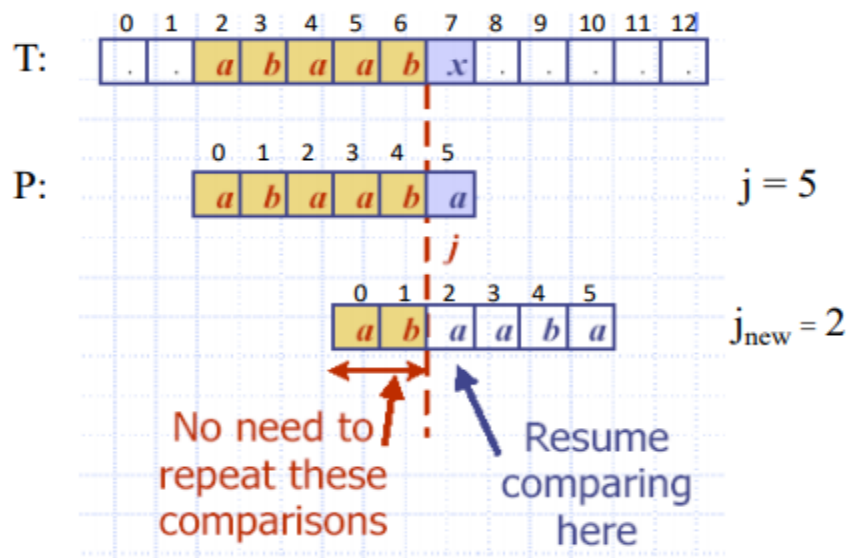
2.1 Algoritma KMP, BM, dan Regex

Pattern matching adalah mencari lokasi pertama di dalam teks yang bersesuaian dengan pattern jika teks tersebut panjang stringnya n karakter dan pattern dengan panjang string m karakter (asumsi $m \ll n$).

A. Algoritma KMP (Knuth-Morris-Pratt)

Algoritma KMP merupakan algoritma yang digunakan untuk melakukan proses pencocokan string (pattern matching) yaitu dengan mencari pattern di dalam teks dengan cara menyisir dari kiri ke kanan seperti algoritma brute force, tetapi algoritma ini lebih pintar daripada brute force.

Untuk menghindari perbandingan yang boros, jika terjadi mismatch antara teks dan pattern P pada $P[j]$ maka perlu dilakukan pencarian prefix dari $P[0..j-1]$ paling besar yang juga merupakan suffix dari $P[1..j-1]$. Inilah yang disebut sebagai border function.



Gambar 2.1.1

B. Algoritma BM(Boyer-Moore)

Algoritma Boyer-Moore adalah algoritma yang mencocokkan karakter dari sebelah kanan pattern. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum.

Cara kerja algoritma Boyer-Moore misal ada sebuah usaha pencocokan yang terjadi pada teks $T[i..i+n-1]$, dan anggap ketidakcocokan pertama terjadi di antara teks $T[i+j]$ dan pattern $P[j]$, dengan $0 < j < n$. Berarti, $T[i+j+1..i+n-1] = P[j+1..n-1]$ dan $a = T[i+j]$ tidak sama dengan $b = P[j]$. Jika u adalah akhiran dari pattern sebelum b dan v adalah sebuah awalan dari pattern, maka penggeseran-penggeseran yang mungkin adalah:

1. Penggeseran good-suffix yang terdiri dari menyejajarkan potongan $T[i+j+1..i+n-1] = P[j+1..n-1]$ dengan kemunculannya paling kanan di pattern yang didahului oleh karakter yang berbeda dengan $P[j]$. Jika tidak ada potongan seperti itu, maka algoritma akan menyejajarkan akhiran v dari $T[i+j+1..i+n-1]$ dengan awalan dari pattern yang sama.
2. Penggeseran bad-character yang terdiri dari menyejajarkan $T[i+j]$ dengan kemunculan paling kanan karakter tersebut di pattern. Bila karakter tersebut tidak ada di pattern, maka pattern akan disejajarkan dengan $T[i+n+1]$.

Secara sistematis, langkah-langkah algoritma Boyer-Moore adalah:

1. Algoritma Boyer-Moore mulai mencocokkan pattern pada awal teks.
2. Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 - 2.1. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).

- 2.2. Semua karakter di pattern cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser pattern dengan memaksimalkan nilai penggeseran good-suffix dan penggeseran bad-character, lalu mengulangi langkah 2 sampai pattern berada di ujung teks.

C. Regular Expression

Regular expression atau biasa disebut regex adalah notasi standar yang mendeskripsikan suatu pola(pattern) berupa urutan karakter atau string. Regex digunakan untuk pencocokan string dengan efisien.

Character Classes

Contract	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

Contract	Deskripsi
----------	-----------

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

Quantifier

Construct	Deskripsi
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

2.2 Chatbot

Chatbot merupakan sebuah layanan obrolan dengan robot/tokoh virtual dengan menggunakan teknologi kecerdasan buatan atau *artificial intelligence* yang menirukan percakapan manusia baik melalui pesan suara maupun pesan tulisan atau keduanya.

Chatbot umumnya digunakan di berbagai industri dengan tujuan untuk menyampaikan informasi. Chatbot juga dapat melakukan berbagai macam tugas sesuai dengan yang sudah diprogram seperti memberi tahu keadaan cuaca, melakukan reservasi sesuatu, atau hanya sekedar untuk hiburan sebagai teman mengobrol.

Cara kerja suatu chatbot adalah dengan memindai kata kunci yang terdapat pada input user, kemudian memberikan respon sesuai kata kunci yang paling cocok atau bisa juga dengan pola kata yang paling mirip. Pengguna akan mengirim pesan atau permintaan dan bots akan mengirimkan respon secara spesifik berdasarkan query yang dikirim. Kemudian bots akan mulai mencari data yang dibutuhkan dari database berdasarkan query yang dikirimkan, lalu setelah data yang dibutuhkan sudah didapatkan bots akan menyampaikan data tersebut melalui tulisan.

BAB III

ANALISIS PEMECAHAN MASALAH

3.1 Langkah Pemecahan Setiap Fitur

Pada tiap fitur yang ada pada chatbot kami, bot akan mengidentifikasi beberapa kata kunci yang sudah tersedia sebelumnya (hardcore), seperti tubes, tucil, kuis, ujian, dan praktikum dan dikombinasikan dengan kata kunci perfitur. Sebagai contoh deadline, sudah, menyelesaikan, diundur, dll. Kata kunci ini nantinya akan menjadi dasar pemrosesan pada program dan database

3.1.1. Fitur Menambahkan suatu task tertentu

Fitur ini memanfaatkan kata kunci berupa jenis task seperti tubes, tucil, ujian, kuis, dan praktikum yang nantinya akan dimasukkan data ke dalam database sesuai kriteria kata kunci tersebut. Pencocokan input pada fitur ini menggunakan algoritma string matching *Knuth Morris Pratt (KMP)* yang akan mengembalikan indeks kata kunci di dalam suatu input, sehingga input kata dalam fitur ini dapat berupa bahasa sehari-hari asalkan memuat kata kunci tersebut. Pengambilan kode kuliah, tanggal, dan deskripsi menggunakan reguler expresion.

3.1.2 Fitur Melihat daftar task yang harus dikerjakan

Fitur ini memanfaatkan kata kunci deadline, yang mana prosesnya terbagi menjadi 3 bagian. Bagian pertama apabila input pengguna berupa deadline interval, seperti “deadline dari tanggal 28/04/2021 hingga 04/0/2021 maka, pengambilan tanggal memanfaatkan reguler expresion sedangkan pengambilan data menggunakan query sql. Bagian kedua adalah apabila input user berupa tenggat deadline, misal “deadline 2 hari kedepan/deadline 4 minggu kedepan” dan memanfaatkan reguler expresion untuk mengekstrak deadline tanggal dan dikonversi tanggalnya sesuai yang dibutuhkan, sedangkan yang terakhir apabila input user berupa permintaan deadline dengan kata kunci tertentu dan tengat waktu tertentu maka akan ditampilkan hasil yang sesuai dengan

query sql. Apabila hasil dari input pengguna tidak ditemukan di database maka akan ditampilkan pesan “tidak ada”

3.1.3 Fitur menampilkan deadline dari suatu task tertentu

Fitur ini memanfaatkan kata kunci deadline dan kata kunci yang dikombinasikan dengan kode mata kuliah. Jika terdapat kata kunci tersebut maka program akan mencari di database sesuai dengan kata kunci dan topik yang sudah diinput oleh user. Pengambilan kode kuliah pada input pengguna menggunakan regular expression. Apabila hasil dari input pengguna tidak ditemukan di database maka akan ditampilkan pesan “tidak ada”

3.1.4 Fitur memperbarui task tertentu

Fitur ini memanfaatkan kata kunci diundur dimana input dari pengguna akan diekstrak berdasarkan kata kunci tersebut dan menggunakan regular expression untuk mendapatkan ID dan tanggal pada input user yang nantinya akan diproses untuk query sql.

3.1.5 Fitur Menandakan suatu task sudah selesai dikerjakan

Fitur ini memanfaatkan kata kunci sudah mengerjakan, atau sudah menyelesaikan, atau udah yang mana akan mengidentifikasi ID dari suatu task yang dimasukkan pengguna menggunakan regular expression. ID tersebut nantinya akan dijadikan query sql untuk mengubah data tersebut.

3.1.6 Fitur menampilkan opsi help

Fitur ini akan menampilkan apa saja daftar task yang bisa dikerjakan oleh bot dan daftar kata kunci.

3.1.7 Fitur menampilkan pesan error apabila bot tidak mengenali input pengguna

Apabila input pengguna tidak mengandung kata kunci suatu task dan tidak mengandung kata kunci tiap fitur maka akan ditampilkan pesan error

3. 2 Fitur Fungsional dan Arsitektur Chatbot

Dalam perangkat lunak yang kami bangun, kami menggunakan bahasa pemrograman Python untuk keperluan backend dan HTML serta CSS untuk pembuatan frontend sedangkan web framework yang kami menggunakan menggunakan flask. Sedangkan untuk penyimpanan data dari perangkat lunak kami memanfaatkan database mysql.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi Teknis Program

1. KMP

```
class KMP:

    def partial(self, pattern):

        result = [0]

        for i in range(1, len(pattern)):

            j = result[i-1]

            while j > 0 and pattern[j] != pattern[i]:

                j = result[j - 1]

            result.append(j + 1 if pattern[j] == pattern[i] else j)

        return result

    def search(self, text, query):

        # Mengembalikan semua query posisi yang cocok dari query
        # pada text input

        partial = self.partial(query)

        ret = []
```

```

j = 0

for i in range(len(text)):
    while j > 0 and text[i] != query[j]:
        j = partial[j - 1]

    if text[i] == query[j]: j += 1

    if j == len(query):
        ret.append(i - (j - 1))

        j = partial[j - 1]

return ret

```

Class KMP ini merupakan implementasi dari algoritma KMP untuk matching pattern pada input user dengan kata kunci yang sudah dibuat. Pada fungsi search nanti akan mengembalikan indeks pada inputan user jika sesuai dengan pattern.

2. inputCommand

```

def inputCommand(text):

    global conn

    global cur

    global turn

    task = ["kuis", "Kuis", "ujian", "Ujian", "Tubes", "tubes",
"Tucil", "tucil", "praktikum", "Praktikum"]

    listText = stemInput2(text)

    output = ""

```

```

# fungsionalitas update

    if "sudah" in listText and "mengerjakan" in listText or "sudah"
in listText and "menyelesaikan" in listText or "udah" in listText
and "ngerjain" in listText:

        output = fungsionalitasDeleteTask(text)


# fungsionalitas help

    elif "help" in listText or "assistant" in listText or "bot" in
listText:

        output = fitur()


# fungsionalitas update task

    elif "diundur" in listText:

        output = fungsionalitasUpdateTask(text)


# fungsioalitas deadline

    elif "deadline" in listText or "Deadline" in listText:

        output = fungsionalitasDeadline(text, task, listText)


# fungsionalitas tambah task

    elif len(checkElmtList(task, listText)) == 1:

        output = fungsionalitasInputTask(text, listText, task)

    turn += 1

```

```

elif "exit" in listText:

    cur.execute("DELETE FROM data;")

    conn.commit()

    return

else:

    output = "Maaf, pesan tidak dikenali<br>"

return output

```

Fungsi inputCommand ini adalah fungsi utama berfungsi memisahkan command-command kata kunci dari input pengguna yang nantinya akan dieksekusi berdasarkan fungsionalitas-fungsionalitasnya. Fungsi ini juga mengembalikan output berupa string yang nantinya dipanggil di bagian frontend untuk ditampilkan.

3. getDate

```

def getDate(date):

#

    result = re.search(r'(\b\d{1,2}\D{0,3})?
(?: (J|j) (?:anuari) | (F|f) (?:ebruari) | (M|m) (?:aret) | (A|a) (?:pril) | (M|m) (?:ei)
| (J|j) (?:uni) | (J|j) (?:uli) | (A|a) (?:ugustus) | (S|S) (?:eptember) | (O|o) (?:ktob
er) | (Nov|Des|nov|des) (?:ember)? ) (?:1\d{3}|2\d{3}) (?:=\D|$)' ,date)

    if result == None:

        reslt =
re.search(r'\b(0?[1-9] | [12] [0-9] | 3[01]) / (0?[1-9] | 1[012]) / (1\d{1} | 2\d{1}) $'
,date)

        if reslt == None:

```



```

        res =
re.search(r'\b(0?[1-9] | [12] [0-9] | 3[01]) / (0?[1-9] | 1[012]) / (1\d{3} | 2\d{3}) $'
, date)

        return res.group()

        return reslt.group()

return result.group()

```

Fungsi getDate ini berguna untuk mengekstrak date dengan format “2 januari 2021” atau “02/01/21” atau format “02/01/2021” dari sebuah input pengguna, algoritma ini memanfaatkan regex yang berupa logika or serta cek kevalidan angka, panjang angka pada setiap tanggal, serta melakukan cek input dengan urutan tertentu.

4. getDualDate

```

def getDualDate(date):

                                result =
re.search(r'\b( (0?[1-9] | [12] [0-9] | 3[01]) / (0?[1-9] | 1[012]) / (1\d{3} | 2\d{3}) \
s\D+ \s(0?[1-9] | [12] [0-9] | 3[01]) / (0?[1-9] | 1[012]) / (1\d{3} | 2\d{3})) $' , date)

    if result == None:

        return result

    result = result.group().split()

    result.pop(1)

    return result

```

Fungsi getDualDate ini berfungsi mengembalikan 2 buah tanggal interval yang akan digunakan dalam fungsionalitas deadline interval dengan format dd/mm/yyyy.

5. getKodeKuliah

```
def getKodeKuliah(text):  
    result = re.search(r'\b(?:IF1\d{3}|IF2\d{3}|IF3\d{3}|IF4\d{3})\b',text)  
    return result
```

Fungsi getKodeKuliah ini berfungsi mengidentifikasi kode kuliah pada input pengguna memanfaatkan logical or pada regex serta kombinasi angka dengan panjang 3.

6. getId

```
def getId(text):  
    result = re.search(r'\b(?:ID: \d{1}|ID: \d{2})\b',text)  
    return result.group()
```

Fungsi getId ini berfungsi untuk mengidentifikasi ID dari input pengguna yang memanfaatkan logical or regex serta kombinasi angka dengan panjang 1 atau 2 dengan regex. Fungsi getId ini nantinya akan digunakan pada pencocokan query di database.

7. getDeadline

```
def getDeadline(date):  
    result = re.search(r'\b(?:((?:[1-4])  
(minggu|bulan|Minggu|Bulan)|(?:([1-9]|[12][0-9]|3[01]) (hari|Hari))))\b',date)  
    return result
```

Fungsi `getDeadline` ini berfungsi untuk mengidentifikasi deadline dengan format suatu n-hari/minggu/bulan ke depan dengan memanfaatkan logical or regex serta kombinasi angka yang valid untuk input hari, minggu, dan bulan.

4.2 Penjelasan Tata Cara Penggunaan Program

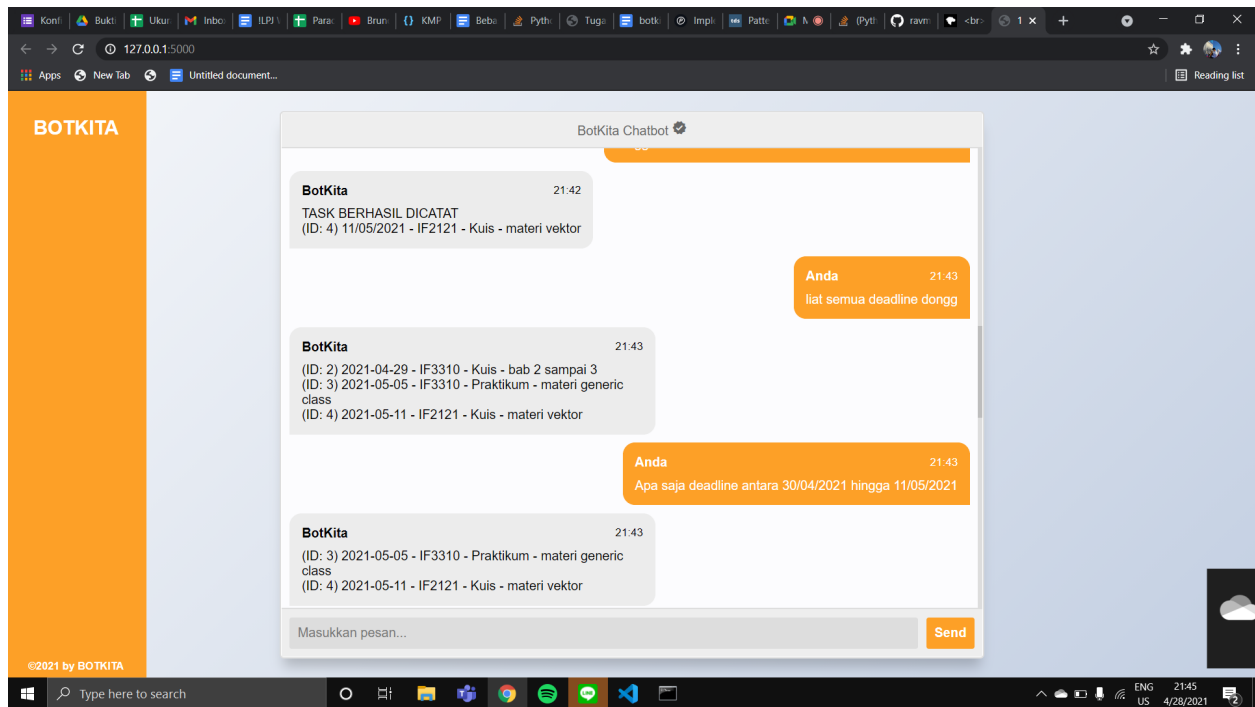
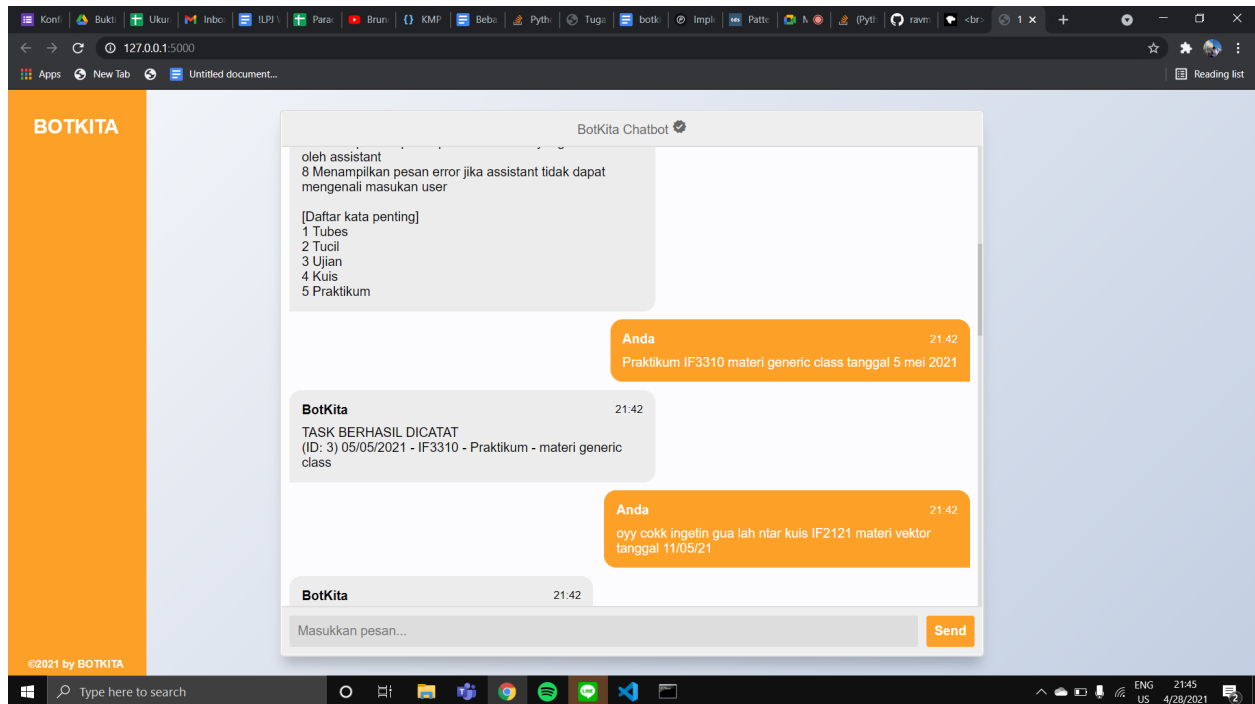
Cara penggunaan program adalah sebagai berikut :

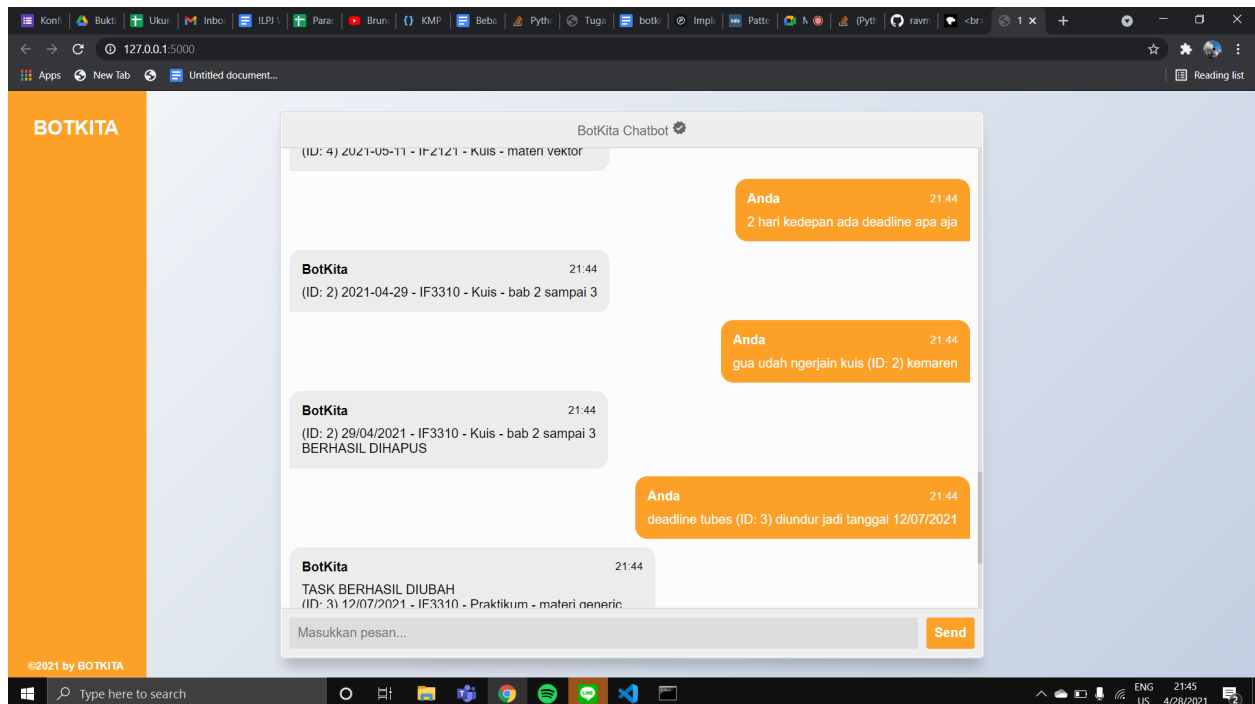
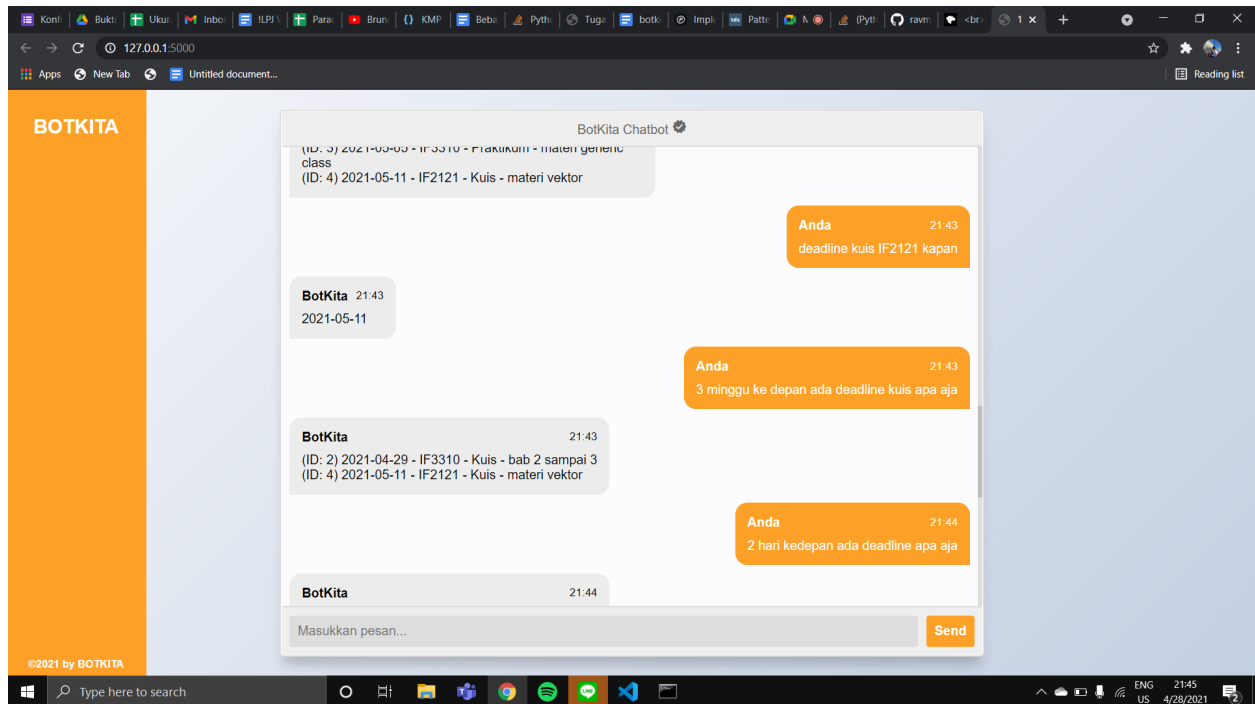
1. Pastikan perangkat sudah terinstal software berikut :
 - a. Python 3.5 atau di atasnya
 - b. PIP 21.1 atau di atasnya
 - c. MariaDB 10.5 atau di atasnya
2. Pastikan perangkat sudah memiliki folder repo 'Tubes3_13519130'
3. Buka MariaDB dan tambahkan database dengan nama `stima3`, kemudian import file `dbChat.sql` pada folder `test` ke database `stima3`.
4. Buka terminal dan arahkan direktori ke folder `src`.
5. Install library flask dengan memasukkan perintah 'pip install Flask'
6. Install library mariadb dengan memasukkan perintah 'pip install mariadb'
7. Jalankan `app.py` dengan perintah 'python app.py'
8. Cli akan menampilkan alamat localhost. Tampilan cli adalah sebagai berikut

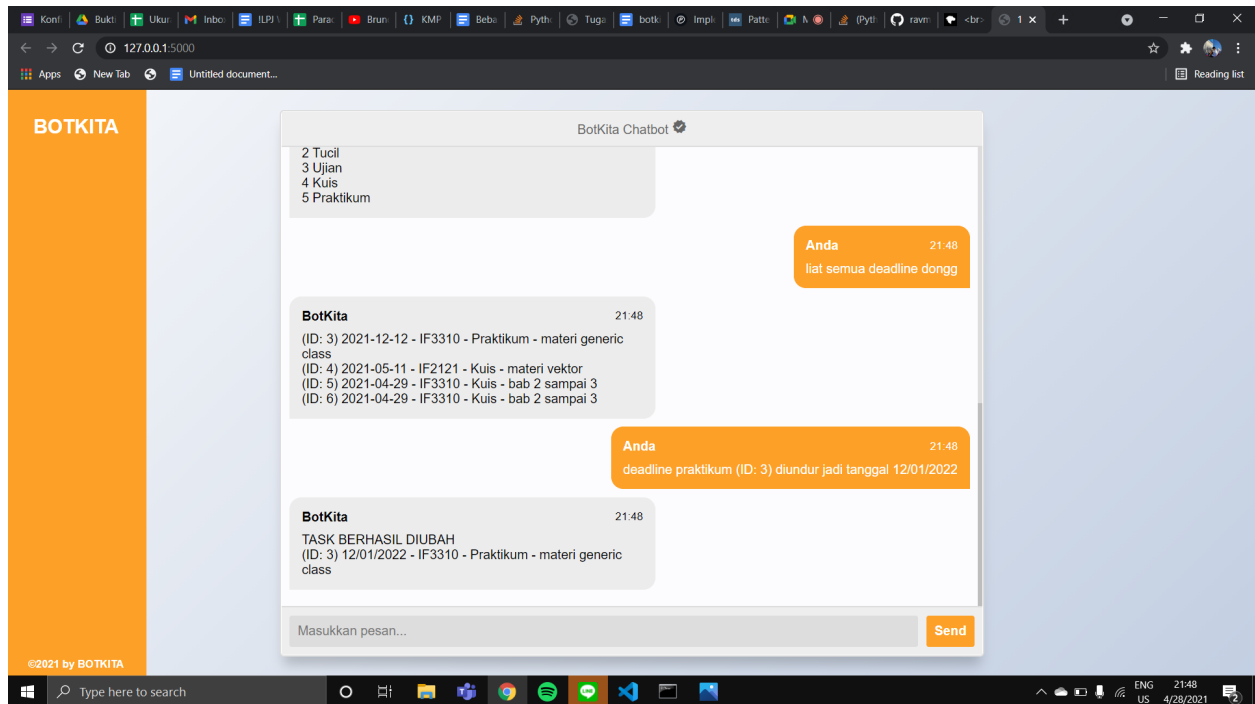
Running on <http://127.0.0.1:5000/> (Press CTRL+C to quit)

9. Buka web browser dan masukkan alamat tersebut.
10. BotKita Chatbot siap digunakan.

4.3 Hasil Pengujian







4. 4 Analisis Hasil Pengujian

Bot dapat menjalankan semua testcase yang diberikan oleh pengguna. Bot dapat menjalankan semua fitur yaitu fitur menambah task baru, melihat daftar task yang harus dikerjakan, menampilkan deadline dari suatu task tertentu, melakukan pembaharuan terhadap suatu task, menandai jika suatu task sudah dilakukan, menampilkan opsi bantuan untuk menampilkan fitur yang dapat digunakan, mendefinisikan list kata penting, dan menampilkan pesan error jika bot tidak mengenali masukan. Bot dapat bekerja dengan query masukan dengan kalimat yang diinginkan pengguna, selama mengandung kata kunci yang dibutuhkan. Bot juga dapat menerima masukan yang diberikan baik dengan bahasa formal maupun bahasa sehari-hari.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Perangkat lunak chatbot kita sudah bisa menyelesaikan semua spek yang ada di laporan dengan kondisional tambahan dapat mengenali bahasa sehari-hari. Selain itu dengan membuat program bot ini, penulis menjadi lebih paham mengenai algoritma Knuth-Morris-Pratt serta penggunaan Regular Expression dalam pencocokan string.

5.2 Saran

Dari eksperimen yang telah kami lakukan, kami memiliki beberapa saran diantaranya lebih disiplin dengan waktu pengerjaan dengan cara tidak menunda-nunda dalam pengerjaannya, karena akan menyulitkan pada saat waktu pengumpulan sebentar lagi. Sebelum pengerjaan tugas ini dilakukan juga disarankan untuk sudah memahami konsep dasar chatbot agar lebih mudah dalam mengerjakan kode program.

5.3 Refleksi

Kalau bisa deadline tugas nya lebih awal agar tidak terlalu keos di akhir dikarenakan banyak deadline tubes mata kuliah lain juga.

DAFTAR PUSTAKA

Munir, Rinaldi. 2006. Diktat Kuliah Strategi Algoritma. Institut Teknologi Bandung: Bandung.

idCloudHost, 2020. Mengenal Apa Itu Chatbot : Definisi, Cara Kerja, Fungsinya. Diakses dari <https://idcloudhost.com/mengenal-apa-itu-chatbot-definisi-cara-kerja-fungsinya/> pada 27 April 2021.