

Tugas Besar 3

IF2211 - Strategi Algoritma



Kelompok 52

Persona

Anggota kelompok :

13519062

Feralezer L. G. Tampubolon

13519103

Bryan Rinaldo

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021

Daftar Isi

Deskripsi tugas	2
Landasan Teori	5
Deskripsi singkat algoritma KMP, BM, dan Regex	5
Penjelasan singkat mengenai Chatbot	5
Analisis Pemecahan Masalah	7
Langkah penyelesaian masalah setiap fitur	7
Fitur fungsional dan arsitektur Chatbot yang dibangun	9
Implementasi dan pengujian	12
Spesifikasi teknis program	12
Penjelasan tata cara penggunaan program	22
Hasil pengujian	24
Analisis hasil pengujian	30
Kesimpulan, saran, dan komentar	31
Daftar Pustaka	32

1. Deskripsi tugas

Dalam tugas besar ini, kami 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:
 - Tanggal (format dibebaskan)
 - Kode Mata Kuliah / Nama Mata Kuliah (dibebaskan)
 - Jenis Tugas (berdasarkan daftar kata penting yang sudah disediakan)
 - Topik Tugas (tidak ada batasan)
- b. Point diatas 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

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
 - Pada periode tertentu (DATE_1 until DATE_2)
Contoh perintah yang dapat digunakan: "Apa saja deadline antara DATE_1 sampai DATE_2?"
 - N minggu ke depan
Contoh perintah yang dapat digunakan: "Deadline N minggu ke depan apa saja?"

- N hari ke depan
Contoh perintah yang dapat digunakan: "Deadline N hari ke depan apa saja?"
 - Hari ini
Contoh perintah yang dapat digunakan: "Apa saja deadline hari ini?"
 - c. Berdasarkan jenis task (kata penting)
 - Sesuai dengan daftar task yang didefinisikan
 - User dapat melihat daftar task dengan jenis task tertentu
 - Misalnya: "3 minggu ke depan ada kuis apa saja?", maka Chatbot akan menampilkan daftar kuis selama 3 minggu kedepan
3. Menampilkan deadline dari suatu task tertentu
 - a. Hanya berlaku untuk task yang bersifat Tugas atau memiliki tenggat waktu
 - b. Misalnya: "Deadline tugas IF2211 itu kapan?"
 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.

2. Landasan Teori

a. Deskripsi singkat algoritma KMP, BM, dan Regex

Algoritma KMP merupakan algoritma yang digunakan untuk melakukan proses pencocokan string. Algoritma ini merupakan jenis Exact String Matching Algorithm yang merupakan pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang sama. Contoh : kata algoritmik akan menunjukkan kecocokan hanya dengan kata algoritmik. Pada algoritma KMP, kita simpan informasi yang digunakan untuk melakukan pergeseran lebih jauh, tidak hanya satu karakter seperti algoritma Brute Force. Algoritma ini melakukan pencocokan dari kiri ke kanan.

Algoritma Boyer-Moore melakukan pencocokan karakter dimulai dari kanan ke kiri. Karakter paling kanan pada pola merupakan karakter pertama yang akan dicocokkan dengan teks. Algoritma ini mempunyai dua fase, yaitu fase preprocessing dan fase pencarian. Pada fase preprocessing terdapat dua buah fungsi untuk menggeser pola ke arah kanan. Kedua fungsi ini disebut good-suffix-shift dan badcharacter-shift. Fungsi good-suffix-shift disimpan ke dalam sebuah tabel bmGs berukuran $m+1$. Sedangkan fungsi bad-character-shift disimpan ke dalam sebuah tabel bmBc yang berukuran n .

Pembentukan tabel bmBc dan bmGs mempunyai kompleksitas waktu $O(m+n)$ dan kompleksitas ruang $O(m+n)$. Sedangkan kompleksitas waktu untuk fase pencarian adalah $O(mn)$. Kasus terbaik untuk algoritma ini mempunyai kompleksitas waktu $O(n/m)$ sedangkan pada kasus terburuk akan terjadi sebanyak $3n$ kali perbandingan untuk pencarian dengan pola yang tidak berulang (periodik).

Regular expression (regex) merupakan sekumpulan notasi dan karakter yang digunakan untuk mendeskripsikan suatu pola pada pencarian berbasis huruf. Dengan regex dimungkinkan untuk mengenali suatu string yang mempunyai karakteristik dan pola tertentu, seperti email, tanggal, nomor kartu kredit, dll. Misal dengan notasi regex $p.+ing$ maka akan menyaring semua kata-kata kecuali kata yang diawali huruf "p" dan mempunyai akhiran ing, seperti playing, praying, pulling, dll.

b. Penjelasan singkat mengenai Chatbot

Istilah Chatbot dengan nama lain chatterbot merupakan sebuah layanan obrolan robot/tokoh virtual dengan kecerdasan buatan atau AI (Artificial Intelligent) yang menirukan percakapan manusia melalui pesan suara, obrolan

teks ataupun keduanya. Singkatnya Chatbot ini merupakan percakapan virtual dimana satu pihak adalah sebuah robot chat yang bertujuan untuk sarana hiburan. Fitur chatbot ini telah digunakan di berbagai industri untuk menyampaikan informasi.

Selain itu Chatbot juga dapat melakukan tugas, seperti pemberitahu cuaca terkini (Weather Bot), kemudian membantu memilih dan memasan bahan makanan (Grocery Bot), membantu melakukan reservasi penerbangan, membantu memberikan solusi atas suatu (Life Advice Bot) dan bot sebagai teman untuk bercakap-cakap seperti SimSimi.

Chatbot menggunakan berbagai teknologi yang mutakhir di dalamnya, seperti Artificial Intelligence (AI), Machine Learning, Deep Learning, dan Natural Language Processing (NLP). Machine Learning dapat di aplikasikan sebagai mesin yang mempelajari, menganalisa dan mengenal bermacam-macam Bahasa pada chatbot, sementara Natural Language Processing (NLP) memiliki kemampuan untuk mengerti dan memahami Bahasa manusia lalu memberikan respon yang sesuai dengan Bahasa yang digunakan pengguna chatbot.

Banyak jenis chatbot yang dapat memberi tanggapan dengan berbagai macam input manusia. Sebenarnya bots ini bekerja dengan memindai kata kunci dalam input, setelah itu dibalas dengan kata kunci yang paling cocok atau bisa pula dengan pola kata yang paling mirip dari basis data tekstual.

Jadi, jika pengguna mengirim suatu permintaan maka bots tersebut akan mengirim kembali respon secara spesifik yang berdasarkan pada query yang dikirim. Pada saat itu yang perlu Anda lakukan hanya memberikannya perintah, selain itu bisa pula membuat suatu percakapan baik text atau audio.

Setelah itu para bots ini mulai mencari data yang dibutuhkan yang berdasarkan kata kunci yang diberikan, baik itu berbentuk text ataupun audio. Setelah mendapatkan data yang dibutuhkan tersebut, akan disampaikan kembali oleh mereka juga dalam bentuk teks maupun audio. Bahkan, tidak hanya itu teknologi ini juga bisa melakukan transaksi yang berdasarkan perintah yang diberikan.

3. Analisis Pemecahan Masalah

a. Langkah penyelesaian masalah setiap fitur

Dalam program Persona, Chatbot yang telah kami buat, hampir seluruh fitur memiliki langkah yang sama. Awalnya kami membuat sebuah fungsi untuk melakukan string matching menggunakan algoritma Knuth–Morris–Pratt dan algoritma Boyer–Moore. Setelah itu kami membuat fungsi string-matching pada kata-kata yang menjadi kunci dari chatbot kami.

Dari input yang dimasukkan oleh pengguna, kami akan melakukan pencarian string matching pada kata-kata yang sudah kami tentukan. Setiap fitur memiliki kata kunci yang berbeda, tetapi ada juga yang memiliki kata kunci yang sama, oleh karena itu kami membuat lebih dari 1 kata kunci untuk setiap fitur yang ada dalam chat bot kami.

Berikut adalah langkah-langkah untuk menyelesaikan :

1. Menambahkan jadwal

Pada penambahan jadwal, kami mencari dari inputan pengguna apakah ada tanggal dengan format dd/mm/yyyy, jenis_tugas dengan kata penting(tucil, tubes, kuis, ujian, praktikum), mata kuliah dengan format (XX0000 (X = huruf 0 = angka)), dan topik dengan format yang berada di dalam sebuah tanda kutip ("topik"). Tetapi tidak hanya itu saja, pada penambahan jadwal agar tidak terjadi ambigu, kami juga tidak menerima kata kunci taskX.

2. Melihat Daftar task

Pada melihat daftar task terdapat berbagai cara untuk melihat daftar task yang sudah di buat. Kita dapat melihat daftar task yang sudah kita buat selama ini, kita dapat melihat daftar task dengan range tanggal yang diminta, kita dapat melihat daftar task beberapa hari ke depan, task dengan beberapa minggu ke depan, bahkan kita bisa melihat daftar task dengan waktu tertentu dengan jenis tugas yang kita mau.

3. Menampilkan deadline dari suatu task tertentu

Pada melihat deadline suatu task tertentu, dari input pengguna terdapat 3 kata kunci yang penting yaitu "deadline", kode mata kuliah, dan jenis tugas yang memiliki deadline seperti tubes atau tucil.

4. Memperbaharui task

Pada pembaharuan task, kata kunci yang penting ada pada tanggal dengan format dd/mm/yyyy dan task X dengan X sebagai id dari sebuah task. Tetapi sebelum itu kami akan melakukan pengecekan apakah task dengan nomor X ada didalam database atau tidak. Jika ada maka dia

akan berhasil untuk memperbarui tanggal, tetapi jika tidak ada maka akan keluar pesan error bahwa task tidak ada.

5. Menghapus task

Pada penghapusan task, kata kunci yang penting ada pada kata "selesai" dan task X dengan X sebagai id dari sebuah task. Tetapi sebelum itu kami akan melakukan pengecekan apakah task dengan nomor X ada didalam database atau tidak. Jika ada maka dia akan berhasil untuk menghapus task, tetapi jika tidak ada maka akan keluar pesan error bahwa task tidak ada.

6. Menampilkan Opsi Help

Pada opsi help ini, pengguna dapat memanggil opsi help dengan 3 kata penting yang bisa dipakai. 3 kata penting yaitu, "persona", "help", dan "fitur". Ketiga kata penting ini tidak harus digunakan secara bersama-sama.

7. Mendefinisikan list kata penting

Pada program yang kami buat, kami menaruh list kata penting pada sebuah file txt yang dipisah dengan sebuah enter atau \n. Adapun list kata penting yang kami buat berbentuk static. List kata penting yang kami buat yaitu [tubes, tucil, ujian, praktikum, kuis]

8. Menampilkan pesan error

Pada chatbot Persona, ketika fitur lain tidak dikenali, maka Persona akan menjawab bahwa pesan yang diberikan tidak dikenali. Kami membuat program agar melakukan cek terlebih dahulu kepada kasus-kasus yang ada, jika tidak ada kasus yang dikenali maka chatbot akan mengeluarkan pesan error.

b. Fitur fungsional dan arsitektur Chatbot yang dibangun

Pada chatbot Persona, kami menyediakan fitur wajib yang terdapat dalam tugas yang diberikan. Terdapat 8 fitur yang masing-masing memiliki fitur berbeda-beda dan tentunya sangat berguna dalam membantu mahasiswa dalam melakukan pencatatan tugas yang diberikan oleh dosen. Berikut adalah fitur-fitur yang terdapat dalam chatbot Persona beserta contoh penggunaannya:

1. Menambahkan Jadwal

Salah satu fitur utama dari chatbot Persona adalah adanya pencatatan jadwal. Persona akan mencatat jadwal ketika pengguna memberikan 4 komponen penting yaitu tanggal, kode mata kuliah, jenis tugas, dan topik tugas.

Contoh dari penggunaan penambahan jadwal ini :

"Persona tolong catat dong, nanti aku bakal ada **Tubes Stima IF2211** dengan topik **"Libur di tahun 2021"** yang dikumpul tanggal **25/12/2021**"

2. Melihat daftar task

Menambahkan jadwal tidak akan berguna jika kita tidak bisa melihat daftar jadwal yang sudah kita catat. Persona akan memperlihatkan daftar deadline sesuai dengan yang diminta sesuai dengan kata kunci yang digunakan. Terdapat 6 cara penggunaan dalam fitur 2 ini :

- Melihat daftar deadline yang telah dicatat

Contoh penggunaan :

"Persona kasih lihat dong, **deadline** apa aja yang udah aku catet"

- Melihat daftar deadline berdasarkan range tanggal

Contoh penggunaan :

"Persona kasih lihat dong, **deadline** apa aja dari tanggal **20/04/2021** sampai tanggal **29/04/2022**"

- Melihat daftar deadline N hari ke depan

Contoh penggunaan :

"Persona kasih lihat dong, **deadline** apa aja **5 hari ke depan**"

- Melihat daftar deadline N minggu ke depan

Contoh penggunaan :

"Persona kasih lihat dong, **deadline** apa aja **2 minggu ke depan**"

- Melihat daftar deadline hari ini

Contoh penggunaan :

"Persona kasih lihat dong, **deadline** apa aja sih **hari ini**"

- Melihat daftar deadline berdasarkan kata penting
Contoh penggunaan :
"Persona tolong cek ada **kuis** apa aja **3 minggu ke depan**"

3. Melihat tanggal/deadline dari sebuah task

Dengan Persona kita juga bisa mencari deadline dari tugas sebuah mata kuliah. Dengan menuliskan 3 kata kunci yaitu "deadline", jenis tugas, dan kode matakuliah maka Persona akan menampilkan tanggal pengumpulan deadline dari tugas yang diminta.

Contoh penggunaan :

"Persona, **deadline tucil IF2211** kapan sih?"

4. Memperbaharui task

Dalam kehidupan nyata walaupun sangat jarang terjadi jika sebuah tugas diundur tetapi mungkin terjadi jika dimajukan , maka kita harus memperbarui tanggal dari pengumpulan tugas tersebut. Oleh karena hal itu, Persona dapat memperbaharui task dengan 2 kata kunci yaitu dengan task X dengan X sebagai ID dari task yang ingin diperbaharui dan tanggal diubah nya tanggal pengumpulan.

Contoh penggunaan :

"Persona deadline **task 2** aku dimajuin nih jadi tanggal **01/01/2022** padahal lagi libur, belum ngerjain apa2 lagi :((("

5. Menghapus task

Ada suatu saat ketika tugas kita akan selesai, walaupun bukan sekarang. Tetapi ketika sudah selesai kita ingin agar task tersebut hilang dari jadwal yang kita buat. Dengan 2 kata penting, Persona dapat membantu untuk membuang task tersebut dari jadwal.

Contoh penggunaan :

"Akhirnya **selesai** juga nih tugas, **task 6** tolong di hapus dong. liat di jadwal aja udah eneg"

6. Menampilkan opsi help

Jika kita belum familiar pada chatbot Persona ini, maka kita butuh sebuah panduan untuk memakai chatbot Persona. Oleh sebab itu kami membuat fitur help dengan 3 kata penting yang tidak harus digunakan secara bersama-sama tetapi tidak boleh ada kata penting lainnya.

Contoh penggunaan :

"Persona aku ga ngerti cara pake kamu :("

7. Mendefinisikan List kata penting

Jika kita ingin menambahkan kata penting, maka kita hanya butuh untuk menambahkannya pada katapenting.txt yang ada pada folder test. Dengan menambahkan katapenting ini, maka kita akan bisa menambahkan jadwal dengan jenis tugas yang berbeda.

8. Menampilkan pesan error

Persona hanya dapat menampilkan pesan yang dikenali oleh nya. Hal ini agar tidak terjadi kasus seperti film "her", jangan sampai ada orang yang jatuh cinta dengan Persona.

Contoh penggunaan :

"kamu cantik banget hari ini, mau ga jadi pacar aku?"

4. Implementasi dan pengujian

a. Spesifikasi teknis program

```
import re
import os
from flask import Flask, render_template, request, redirect, url_for,
abort, send_from_directory
import sqlite3
import datetime
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.sql.expression import and_

def findFails(pattern):
    fails = []
    fails.append(0)
    i = 1
    j = 0
    while i < len(pattern):
        if pattern[j] == pattern[i]:
            fails.append(j + 1)
            i += 1
            j += 1
        elif j > 0:
            j = fails[j - 1]
        else:
            fails.append(0)
            i += 1
    return fails

def knuthMorrisPratt(string, pattern):
    fails = findFails(pattern)
    i = 0
    j = 0
    while (i < len(string)):
        if string[i] == pattern[j]:
            if j == (len(pattern) - 1):
                return i - len(pattern) + 1
            i += 1
            j += 1
        elif j > 0:
```

```

        j = fails[j - 1]
    else:
        i += 1
    return -1

def findLastOccurence(pattern):
    lastOccurence = {}
    for i in range(len(pattern)):
        lastOccurence[pattern[i]] = i
    return lastOccurence

def boyerMoore(string, pattern):
    lastOccurence = findLastOccurence(pattern)
    i = len(pattern) - 1
    if i > (len(string) - 1):
        return -1
    j = len(pattern) - 1
    do = True
    while do or (i <= (len(string) - 1)):
        if string[i] == pattern[j]:
            if j == 0:
                return i
            else:
                i -= 1
                j -= 1
        else:
            if string[i] in lastOccurence:
                i = i + len(pattern) - min(j, (1 + lastOccurence[string[i]]))
                j = len(pattern) - 1
            else:
                i = i + len(pattern) - min(j, 0)
                j = len(pattern) - 1
        if do:
            do = False
    return -1

def regexTanggal(s):
    tanggals = re.findall('(\b\d{2}/([1-9]|1[0-2]|0[1-9])/\d{4}\b)', s)
    tanggal = []
    for t in tanggals:

```

```

        td = datetime.datetime.strptime(t[0], "%d/%m/%Y")
        tanggal.append(td)
    return tanggal

def regexKataPenting(s):
    katapenting = []
    with open(os.path.join(app.config["UPLOAD_PATH"], "katapenting.txt"),
"r") as filekata:
        for line in filekata:
            katapenting.append(line.rstrip())
    x = '\\b'+'|'.join(katapenting)+'\\b'
    katapentingreg = re.findall(x , s.title())
    return katapentingreg

def regexMatkul(s):
    matkul = re.findall('(\\b[a-zA-Z][a-zA-Z]\\d{4}\\b)', s)
    return matkul

def regexTopik(s):
    topik = re.findall(r'"(.+?)"', s)
    return topik

def kmpDeadline(s):
    return knuthMorrisPratt(s.lower(), "deadline")

def regexNMingguKeDepan(s):
    minggu = re.findall('(\\-?[0-9]+ [mM]inggu [kK]e [dD]epan\\b)', s)
    return minggu

def regexNHariKeDepan(s):
    hariN = re.findall('(\\-?[0-9]+ [hH]ari [kK]e [dD]epan\\b)', s)
    return hariN

def regexHariIni(s):
    hariIni = re.findall('(\\b[hH]ari [iI]ni\\b)', s)
    return hariIni

def regexTaskX(s):
    taskX = re.findall('(\\b[tT]ask [0-9]+\\b)', s)
    return taskX

```

```

def kmpSelesai(s):
    return knuthMorrisPratt(s.lower(), "selesai")

def bmPersona(s):
    return boyerMoore(s.lower(), "persona")

def bmFitur(s):
    return boyerMoore(s.lower(), "fitur")

def bmHelp(s):
    return boyerMoore(s.lower(), "help")

def pesanTidakDikenali():
    return ["Maaf, pesan tidak dikenali"]

app = Flask(__name__)
app.config["UPLOAD_PATH"] = "../test"
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///'+
os.path.join(app.config["UPLOAD_PATH"], "jadwal.db")
app.config['SQLALCHEMY_TRACK_MODIFICATION'] = False
db = SQLAlchemy(app)

class Jadwal(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    tanggal = db.Column(db.DateTime, nullable = False)
    matkul = db.Column(db.String(255), nullable = False)
    jenis_tugas = db.Column(db.String(255), nullable = False)
    topik_tugas = db.Column(db.String(255), nullable = False)

def addJadwal(tanggal, matkul, jenis_tugas, topik_tugas):
    jadwal = Jadwal(tanggal=tanggal, matkul=matkul.upper(),
jenis_tugas=jenis_tugas.title(), topik_tugas=topik_tugas)
    db.session.add(jadwal)
    db.session.flush()
    db.session.commit()
    return jadwal.id

@app.route("/")
@app.route("/index")

```



```

def index():
    return render_template("index.html")

@app.route("/chat", methods=["GET"])
def chat():
    query = request.args.get("q")

    # lakukan semua pengecekan string
    tanggal = regexTanggal(query)
    kataPenting = regexKataPenting(query)
    matkul = regexMatkul(query)
    topik = regexTopik(query)
    deadline = kmpDeadline(query)
    nMingguKeDepan = regexNMingguKeDepan(query)
    nHariKeDepan = regexNHariKeDepan(query)
    hariIni = regexHariIni(query)
    taskX = regexTaskX(query)
    selesai = kmpSelesai(query)
    persona = bmPersona(query)
    fitur = bmFitur(query)
    fiturHelp = bmHelp(query)

    response = pesanTidakDikenali()
    # Kasus 1: menambahkan jadwal ke database
    if (len(tanggal) == 1) and (len(matkul) == 1) and (len(kataPenting) ==
1) and (len(topik) == 1) and (len(taskX) == 0) and (fiturHelp == -1):
        itemid = addJadwal(tanggal[0], matkul[0], kataPenting[0], topik[0])
        if itemid > 0:
            response = ["[TASK BERHASIL DICATAT]", "(ID: " + str(itemid) + ") "
+ tanggal[0].strftime('%d/%m/%Y') + " - " + matkul[0].upper() + " - " +
kataPenting[0].title() + " - " + topik[0]]
        elif (len(tanggal) <= 2) and (len(matkul) == 0) and (len(kataPenting) ==
0) and (len(topik) == 0) and (deadline > -1) and (len(taskX) == 0) and
(fiturHelp == -1):
            if (len(tanggal) == 0) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 0):
                deadlines = Jadwal.query.all()
                if len(deadlines) == 0:
                    response = ["Tidak ada"]
            else:

```

```

        response = ["[Daftar Deadline]"]
        i = 1
        for dl in deadlines:
            response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
            i += 1
        elif (len(tanggal) == 2) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 0):
            deadlines =
Jadwal.query.filter(Jadwal.tanggal.between(min(tanggal[0], tanggal[1]),
max(tanggal[0], tanggal[1]))).all()
            if len(deadlines) == 0:
                response = ["Tidak ada"]
            else:
                response = ["[Daftar Deadline]"]
                i = 1
                for dl in deadlines:
                    response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
                    i += 1
                elif (len(tanggal) == 0) and (len(nMingguKeDepan) == 1) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 0):
                    todayDate = datetime.date.today()
                    weeks = nMingguKeDepan[0].split(" ")
                    deadlines = Jadwal.query.filter(Jadwal.tanggal.between(todayDate,
(todayDate + datetime.timedelta(days=1, weeks=int(weeks[0])))).all()
                    if len(deadlines) == 0:
                        response = ["Tidak ada"]
                    else:
                        response = ["[Daftar Deadline]"]
                        i = 1
                        for dl in deadlines:
                            response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
                            i += 1
                        elif (len(tanggal) == 0) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 1) and (len(hariIni) == 0):

```

```

todayDate = datetime.date.today()
days = nHariKeDepan[0].split(" ")
deadlines = Jadwal.query.filter(Jadwal.tanggal.between(todayDate,
(todayDate + datetime.timedelta(days=(int(days[0]) + 1))))).all()
if len(deadlines) == 0:
    response = ["Tidak ada"]
else:
    response = ["[Daftar Deadline]"]
    i = 1
    for dl in deadlines:
        response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
        i += 1
    elif (len(tanggal) == 0) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 1):
        todayDate = datetime.date.today()
        tomorrowDate = todayDate + datetime.timedelta(days=1)
        deadlines = Jadwal.query.filter(Jadwal.tanggal.between(todayDate,
tomorrowDate)).all()
        if len(deadlines) == 0:
            response = ["Tidak ada"]
        else:
            response = ["[Daftar Deadline]"]
            i = 1
            for dl in deadlines:
                response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
                i += 1
            elif (len(tanggal) <= 2) and (len(matkul) == 0) and (len(kataPenting) ==
1) and (len(topik) == 0) and (deadline >= -1) and (len(taskX) == 0) and
(fiturHelp == -1):
                if (len(tanggal) == 0) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 0):
                    deadlines =
Jadwal.query.filter_by(jenis_tugas=kataPenting[0].title()).all()
                if len(deadlines) == 0:
                    response = ["Tidak ada"]
                else:

```

```

        response = ["[Daftar Deadline]"]
        i = 1
        for dl in deadlines:
            response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
            i += 1
        elif (len(tanggal) == 2) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 0):
            deadlines =
Jadwal.query.filter(and_(Jadwal.tanggal.between(min(tanggal[0],
tanggal[1]), max(tanggal[0], tanggal[1])), (Jadwal.jenis_tugas ==
kataPenting[0].title()))).all()
            if len(deadlines) == 0:
                response = ["Tidak ada"]
            else:
                response = ["[Daftar Deadline]"]
                i = 1
                for dl in deadlines:
                    response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
                    i += 1
                elif (len(tanggal) == 0) and (len(nMingguKeDepan) == 1) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 0):
                    todayDate = datetime.date.today()
                    weeks = nMingguKeDepan[0].split(" ")
                    deadlines =
Jadwal.query.filter(and_(Jadwal.tanggal.between(todayDate, (todayDate +
datetime.timedelta(days=1, weeks=int(weeks[0])))), (Jadwal.jenis_tugas ==
kataPenting[0].title()))).all()
                    if len(deadlines) == 0:
                        response = ["Tidak ada"]
                    else:
                        response = ["[Daftar Deadline]"]
                        i = 1
                        for dl in deadlines:
                            response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)

```

```

        i += 1
        elif (len(tanggal) == 0) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 1) and (len(hariIni) == 0):
            todayDate = datetime.date.today()
            days = nHariKeDepan[0].split(" ")

            deadlines =
Jadwal.query.filter(and_(Jadwal.tanggal.between(todayDate, (todayDate +
datetime.timedelta(days=(int(days[0]) + 1))), (Jadwal.jenis_tugas ==
kataPenting[0].title()))).all()
            if len(deadlines) == 0:
                response = ["Tidak ada"]
            else:
                response = ["[Daftar Deadline]"]
                i = 1
                for dl in deadlines:
                    response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
                    i += 1
        elif (len(tanggal) == 0) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 1):
            todayDate = datetime.date.today()
            tomorrowDate = todayDate + datetime.timedelta(days=1)

            deadlines =
Jadwal.query.filter(and_(Jadwal.tanggal.between(todayDate, tomorrowDate),
(Jadwal.jenis_tugas == kataPenting[0].title()))).all()
            if len(deadlines) == 0:
                response = ["Tidak ada"]
            else:
                response = ["[Daftar Deadline]"]
                i = 1
                for dl in deadlines:
                    response.append(str(i) + ". (ID: " + str(dl.id) + ") " +
dl.tanggal.strftime('%d/%m/%Y') + " - " + dl.matkul + " - " +
dl.jenis_tugas + " - " + dl.topik_tugas)
                    i += 1
        elif(len(matkul)==1) and (deadline>-1) and (len(kataPenting)==1) and
(fiturHelp == -1):
            deadlines = Jadwal.query.filter(and_(Jadwal.matkul==matkul[0].upper(),
Jadwal.jenis_tugas == kataPenting[0].title()))).all()

```

```

        if(kataPenting[0].title() != "Tucil" and kataPenting[0].title()
!="Tubes"):
            response = ["Bukan deadline kak :("]
        else:
            if deadlines:
                response = ["[Daftar Tanggal Deadline]"]
                i = 1
                for dl in deadlines:
                    response.append(dl.tanggal.strftime('%d/%m/%Y'))
                    i += 1
            else:
                response = ["Yey, gaada deadline"]
            #KASUS 4 = UPDATE TANGGAL (kata kunci = tanggal dan task X)
        elif (len(tanggal) == 1) and (len(taskX)==1) and (fiturHelp == -1):
            taskList = taskX[0].split(" ")
            idDelete = int(taskList[1])
            update_tanggal = Jadwal.query.filter(Jadwal.id == idDelete).first()
            if update_tanggal:
                update_tanggal.tanggal = tanggal[0]
                db.session.commit()
                response = ["[TASK BERHASIL DIUPDATE]"]
            else:
                response = ["[TASK TIDAK ADA TOLONG CEK KEMBALI]"]
            #KASUS 5 = HAPUS TANGGAL (kata kunci = task X dan selesai)
        elif (len(tanggal) == 0) and (len(taskX)==1) and (selesai>-1) and
(fiturHelp == -1):
            taskList = taskX[0].split(" ")
            idDelete = int(taskList[1])
            delete_jadwal = Jadwal.query.filter(Jadwal.id == idDelete).first()
            if delete_jadwal:
                db.session.delete(delete_jadwal)
                db.session.commit()
                response = ["[TASK BERHASIL DIHAPUS]"]
            else:
                response = ["[TASK TIDAK ADA TOLONG CEK KEMBALI]"]
        elif ((persona > -1) or (fitur > -1) or (fiturHelp > -1)) and
(len(tanggal) == 0) and (len(kataPenting) == 0) and (len(matkul) == 0) and
(len(topik) == 0) and (deadline == -1) and (len(nMingguKeDepan) == 0) and
(len(nHariKeDepan) == 0) and (len(hariIni) == 0) and (len(taskX) == 0) and
(selesai == -1):

```

```

response = ["[Fitur]"]

with
open(os.path.join(os.path.dirname(os.path.abspath(os.path.realpath(__file_
_))), "static/fitur.txt"), "r", encoding="utf-8") as fiturFile:
    for line in fiturFile:
        response.append(line)
response.append("[Daftar kata penting]")
with open(os.path.join(app.config["UPLOAD_PATH"], "katapenting.txt"),
"r") as kataPentingFile:
    kataPentingIdx = 1
    for line in kataPentingFile:
        response.append(str(kataPentingIdx) + ". " + line)
        kataPentingIdx += 1

return render_template("chat.html", query=query, response=response)

```

b. Penjelasan tata cara penggunaan program

Terdapat beberapa fitur yang ada pada chatbot Persona. Berikut adalah tata cara penggunaan chatbot Persona.

Kata kunci :

- Kode mata kuliah (XX0000 (X huruf dan 0 sebuah angka))
- Jenis tugas (tubes ,tucil, ujian, praktikum, kuis)
- Tanggal (dd/mm/yyyy)
- Topik kuliah ("topik" (diapit dengan tanda petik))
- Deadline ("deadline")
- N minggu ke depan ("N minggu ke depan" (N angka))
- N hari ke depan ("N hari ke depan" (N angka))
- Hari ini ("hari ini")\
- Task x ("task X" (X sebuah angka))
- Selesai ("selesai")
- Persona ("persona")
- fitur("fitur")

1. Menambahkan jadwal

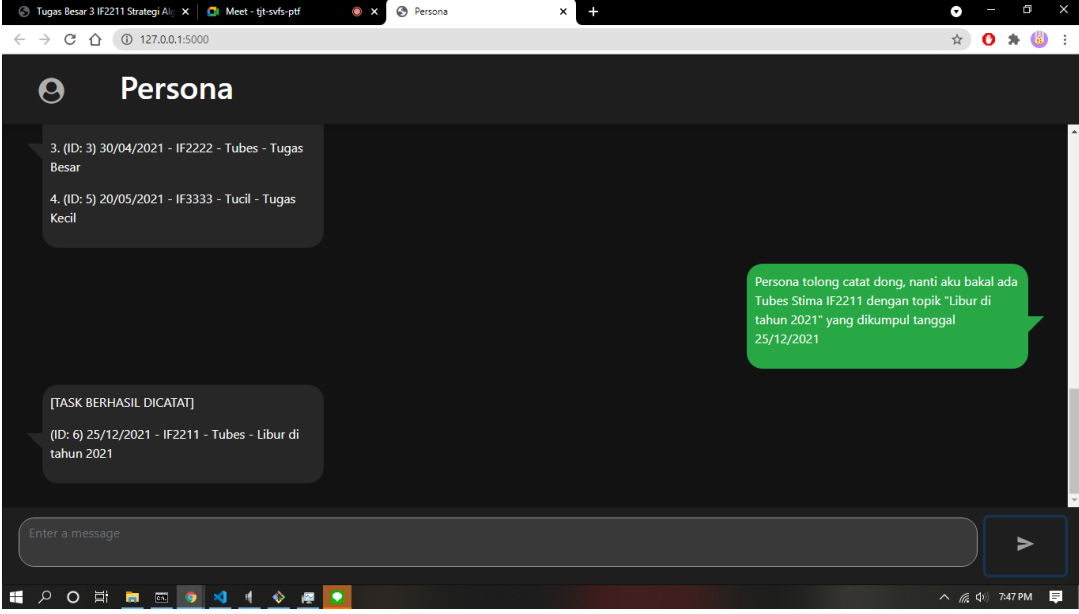
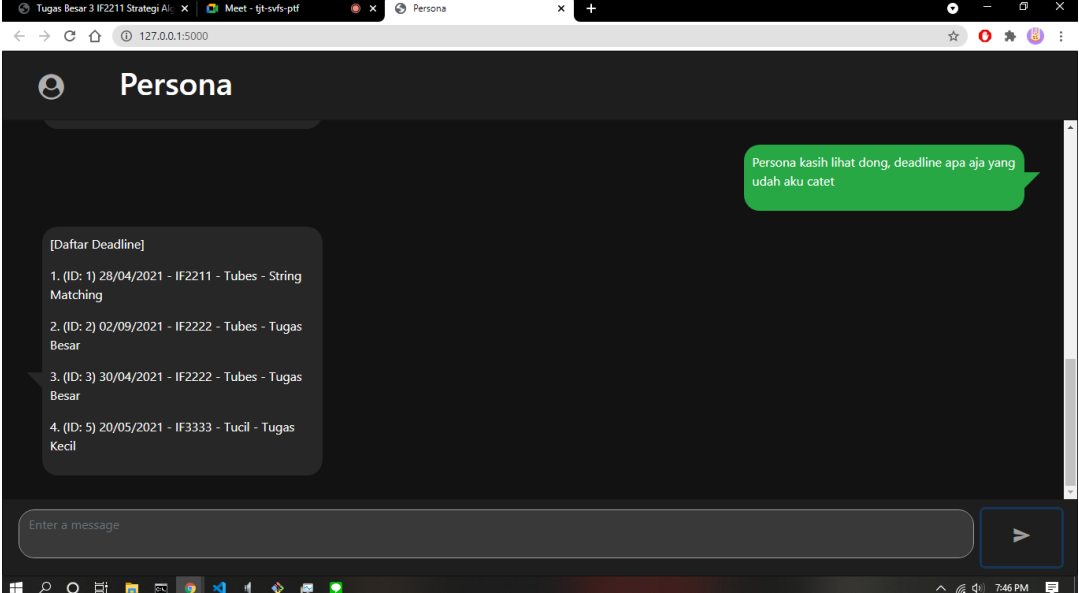
Membutuhkan 4 kata kunci: kode mata kuliah, jenis tugas, tanggal, topik kuliah.

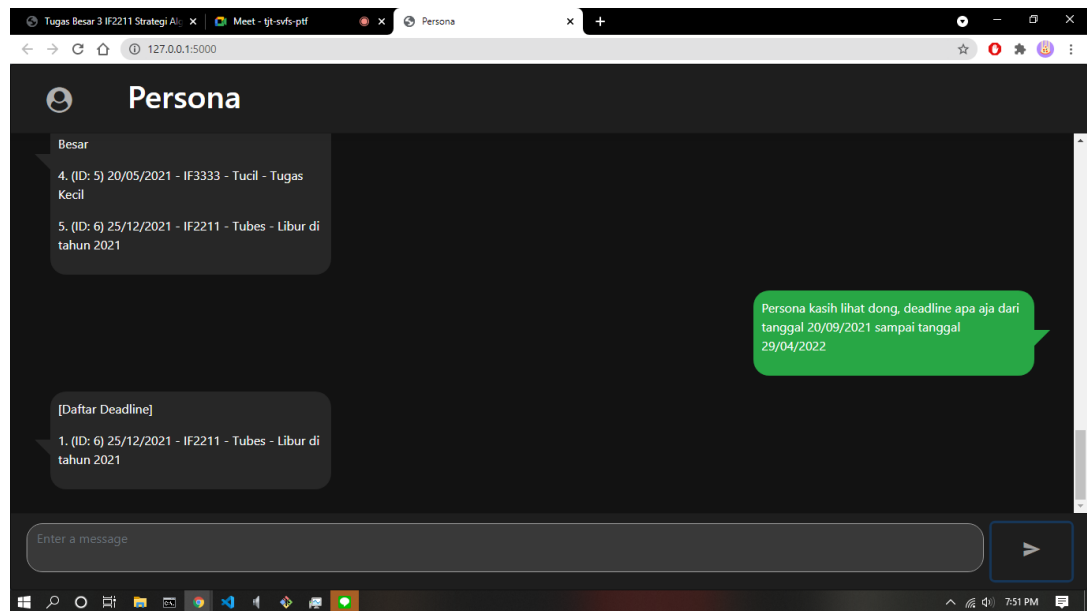
2. Melihat daftar task

- Membutuhkan 1 kata kunci: deadline
- Membutuhkan 3 kata kunci: deadline dan 2 tanggal

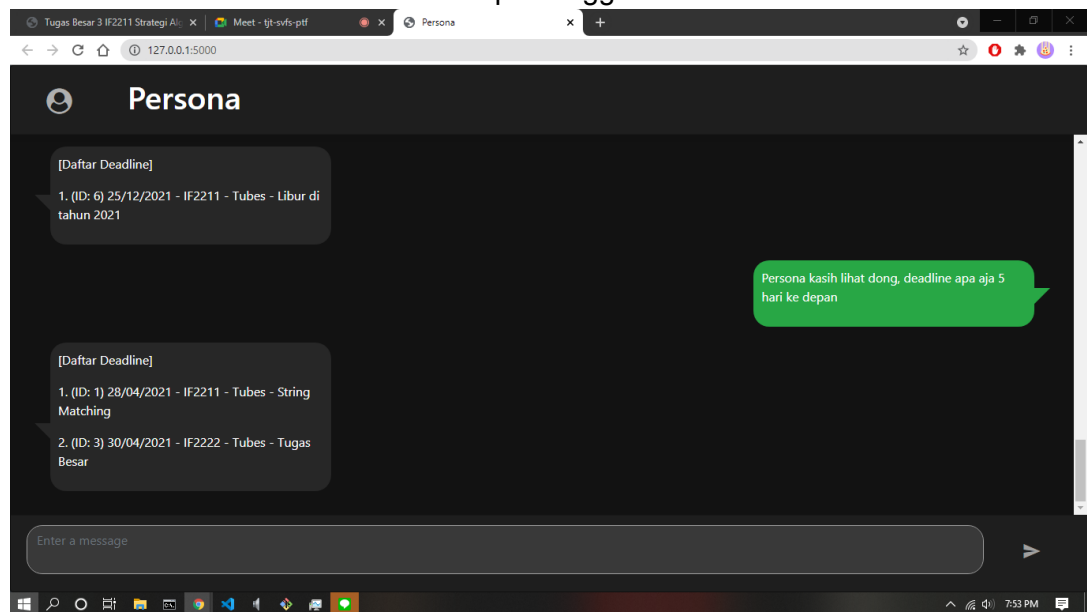
- Membutuhkan 2 kata kunci: deadline dan N minggu ke depan
 - Membutuhkan 2 kata kunci: deadline dan N hari ke depan
 - Membutuhkan 2 kata kunci: deadline dan hari ini
 - Membutuhkan 2 kata kunci: jenis tugas dan range hari kedepan
3. Melihat deadline task
Membutuhkan 3 kata kunci: deadline, kode mata kuliah, dan jenis tugas yang terdapat deadline(tubes dan tucil)
 4. Update task
Membutuhkan 2 kata kunci: tanggal dan task X
 5. Hapus task
Membutuhkan 2 kata kunci: task X dan selesai
 6. Menampilkan help
Membutuhkan 1 kata kunci diantara 3 kata kunci: persona, help, dan fitur.

c. Hasil pengujian

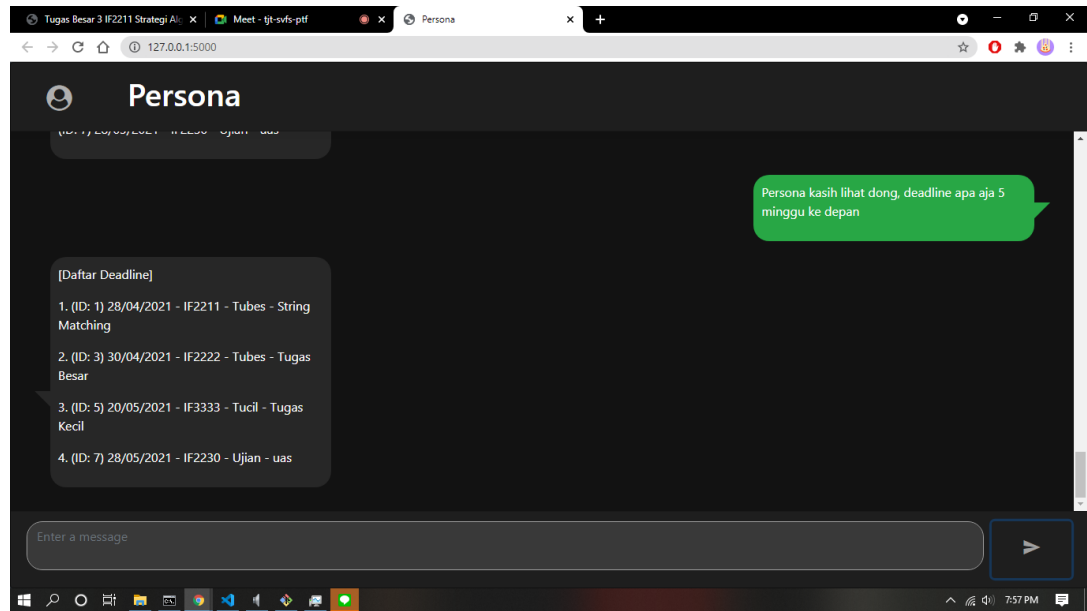
No.	Fitur	Screenshot Hasil pengujian
1.	Menambahkan jadwal	 <p>Mencatat tubes IF2211 untuk tanggal 25/12/2021 dengan topik “libur di tahun 2021”</p>
2.	Melihat daftar task	 <p>Melihat deadline apa saja</p>



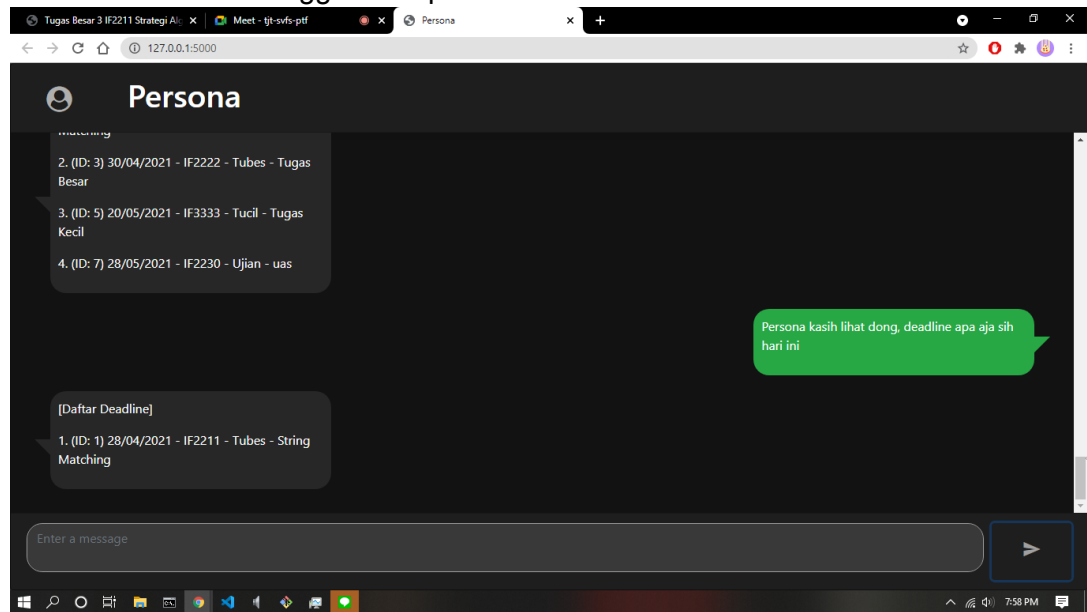
Melihat deadline dari 20/09/2021 sampai tanggal 29/04/2022



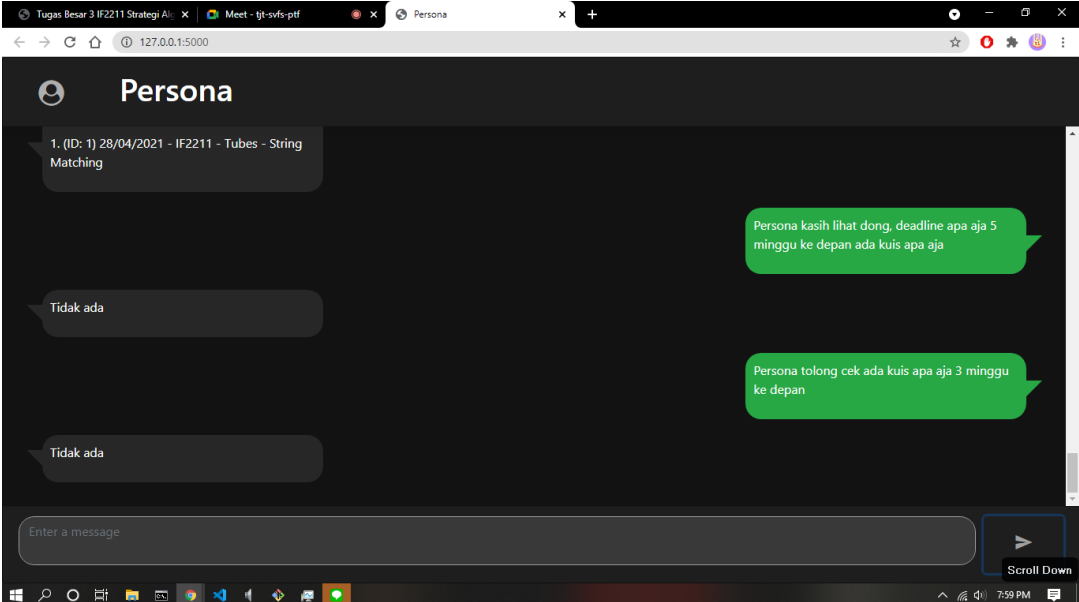
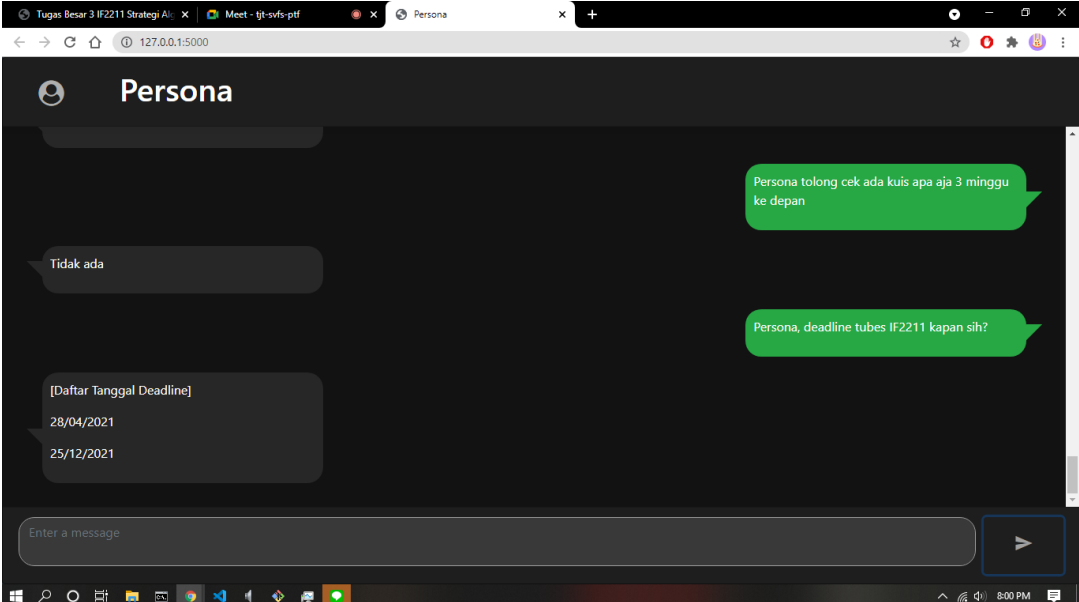
Melihat deadline 5 hari ke depan

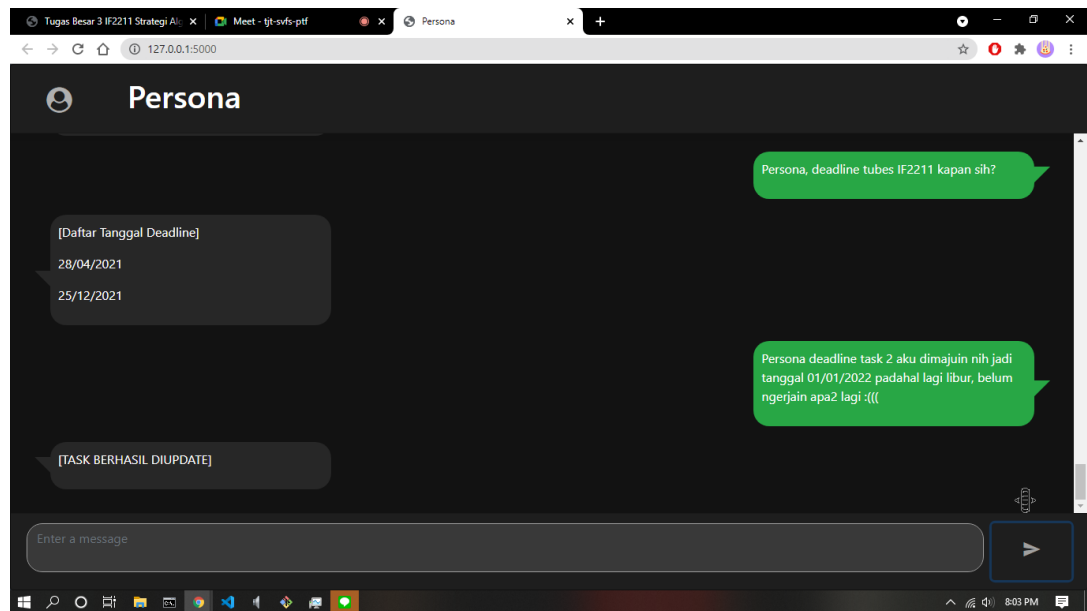


Melihat deadline 5 minggu ke depan

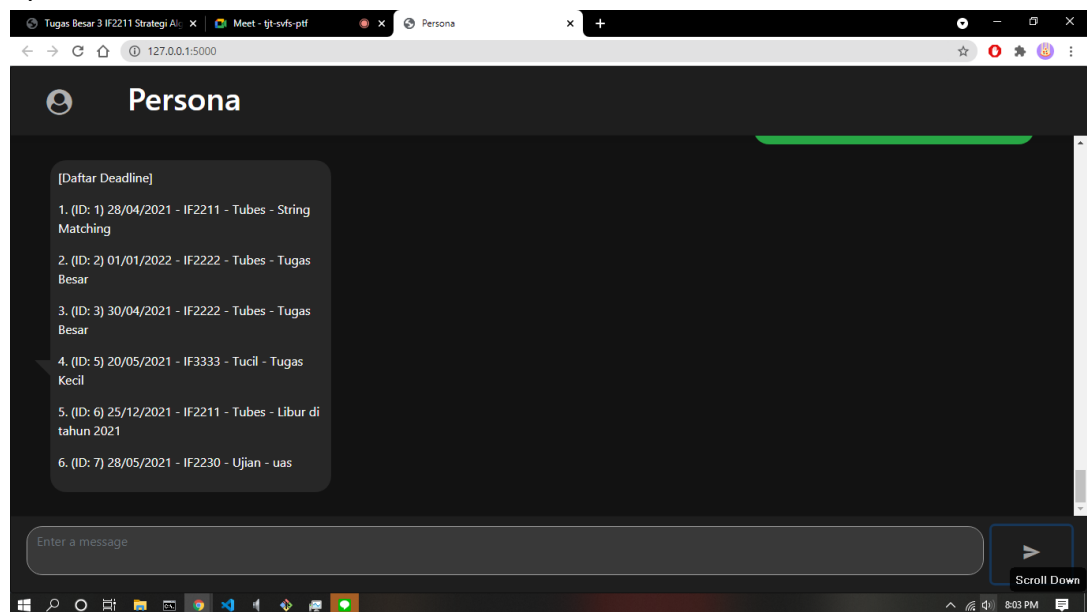


Melihat deadline hari ini

	 <p>Melihat kuis 3 minggu ke depan</p>
3.	<p>Melihat deadline task</p>  <p>Melihat deadline tubes IF2211</p>
4.	<p>Update task</p>

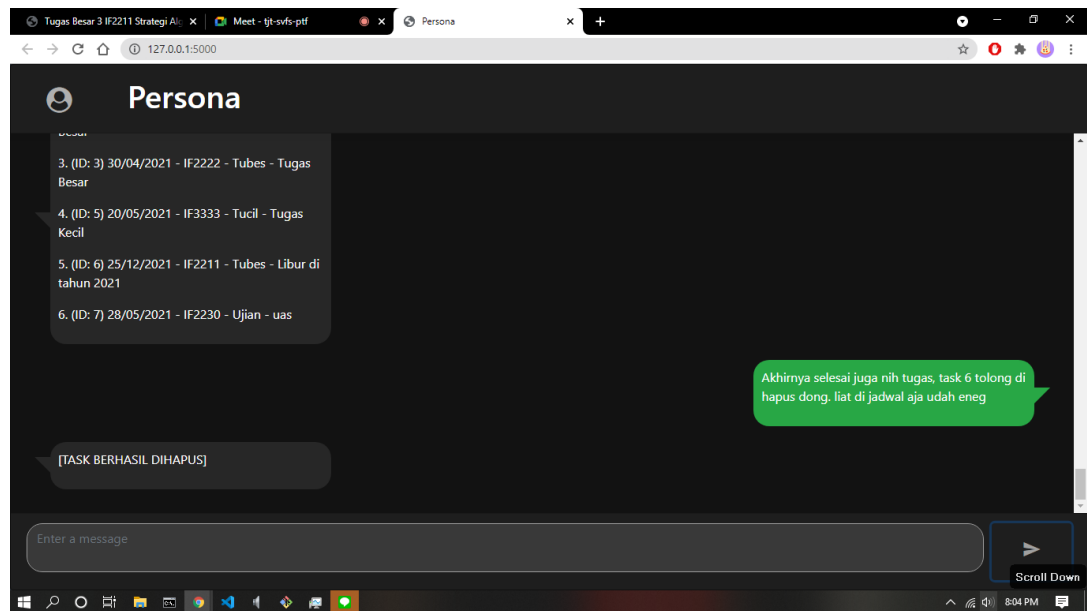


Update berhasil

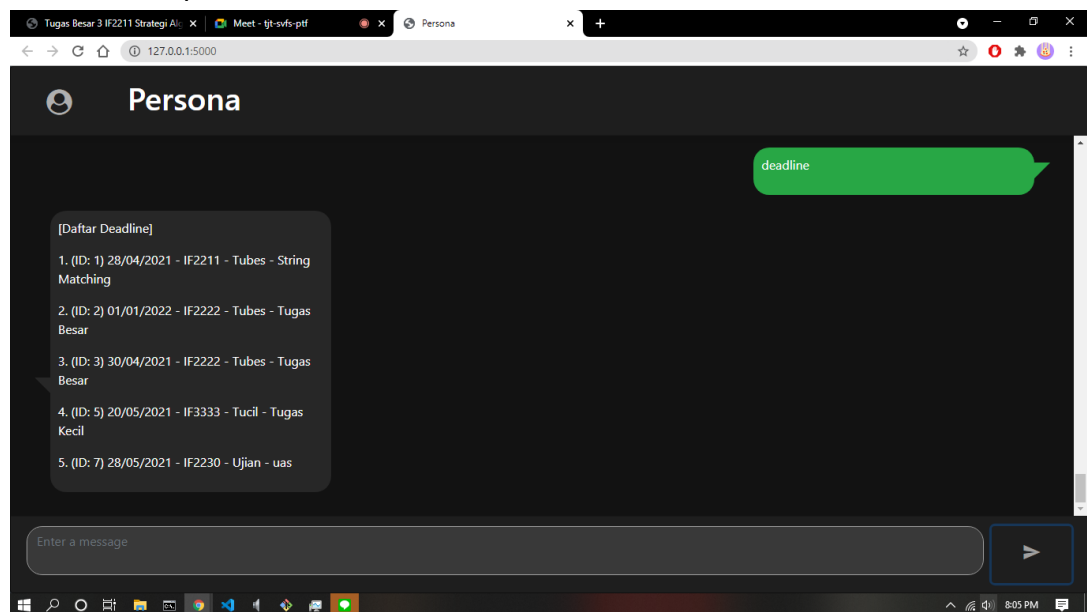


Bukti terupdate

5. **Hapus task**

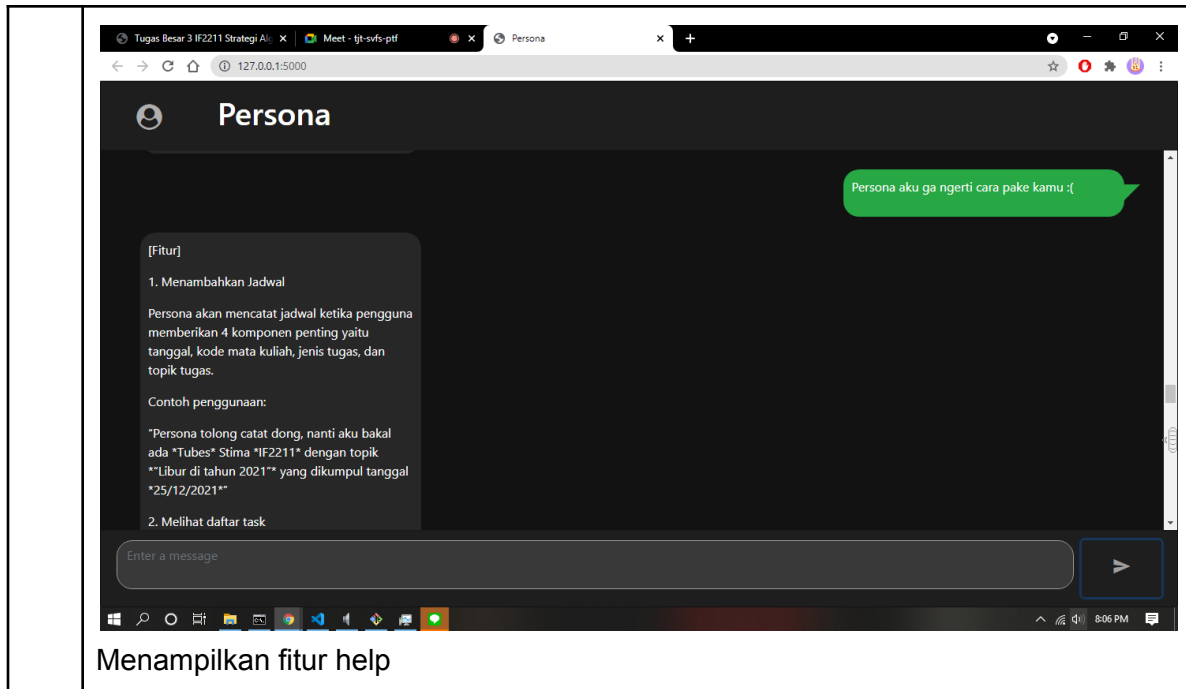


Berhasil di hapus



Bukti terhapus

6. **Menampilkan help**



d. Analisis hasil pengujian

Berdasarkan screenshot hasil pengujian, terlihat bahwa persona berhasil melakukan string matching/pattern matching pada query user dan menjalankan fungsi yang dimaksud oleh query user tersebut. Persona berhasil menjalankan request penambahan task baru ke dalam daftar deadline. Persona juga berhasil melihat daftar deadline, baik dengan maupun tanpa filter. Persona juga berhasil melakukan update dan penghapusan task dari daftar deadline. Terakhir, persona juga dapat menampilkan help/daftar fitur yang ia miliki.

5. Kesimpulan, saran, dan komentar

Setelah kurang lebih 2 minggu diberikan tugas untuk membuat chatbot pintar agar bisa mencatat jadwal yang kita berikan, kami berhasil untuk membuat chatbot tersebut. Chatbot yang kami beri nama Persona memiliki 8 fitur yang sama dengan yang diberikan pada spesifikasi tugas.

Kami kesulitan karena banyak nya tugas yang diberikan oleh dosen, sehingga waktu untuk mengerjakan jadi kurang dan sangat sulit untuk membagi waktu antar mata kuliah. Bukan hanya berhenti disana, awalnya kami juga sangat bingung dengan bagaimana cara memulai buat chatbot. Tetapi dengan melihat youtube dan belajar dari internet, kami mampu untuk mengerjakan bahkan untuk menyelesaikan tugas ini.

Dari tugas besar 3 Strategi Algoritma ini, kami belajar banyak hal. Bukan hanya belajar bahasa python flask yang kami gunakan, tetapi kami juga dapat mempelajari strategi algoritma. Kita bisa tau bahwa terdapat algoritma yang dapat membantu kita mencari sebuah string yang dibutuhkan. Bukan berhenti disana tetapi kita bisa membuat sebuah chatbot dari string matching tersebut.

Bukan hanya belajar hard-skill, tetapi kita juga belajar soft-skill yang tidak bisa kita dapatkan selain dari pengalaman mengerjakan tugas ini. Kita bisa belajar bagaimana cara bekerja sama, belajar bagaimana cara berkomunikasi, dan yang paling penting bagaimana cara mengatur waktu di kehidupan mahasiswa Informatika yang tidak lepas dengan banyaknya tugas besar ditambah lagi akhir semester yang sudah pasti lebih banyak lagi tugasnya.

Untuk saran, menurut kami pemilihan topik tugas ini sudah sangat baik. Proporsi pembagian waktu sudah sangat pas dan tugas yang diberikan juga tidak berlebihan. Topik yang diambil untuk tugas besar ini juga sangat menarik. Topik ini bisa menjadi relevan terhadap beberapa dari kita di masa depan. Topik yang menarik ini membuat para mahasiswa lebih semangat dalam mengerjakan tugas besar ini. Tetapi karena sudah terdapat 3 tugas besar, menurut kami 3 tugas besar terlalu banyak belum lagi ditambah makalah di akhir semester. Mungkin sebaiknya, tugas makalah ditiadakan karena tugas besar seperti ini lebih relevan dengan anak informatika.

Daftar Pustaka

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2005-2006/Makalah2006/MakalahStmik2006-07.pdf>
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/Makalah/MakalahStmik03.pdf>
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>