

LAPORAN TUGAS BESAR III IF2211

STRATEGI ALGORITMA

Semester II Tahun 2020/2021



Oleh
Kelompok LazyBot

Muhammad Rifky Muthahhari	13519016
Aurelius Marcel Candra	13519198
Kevin Katsura Dani Sitanggang	13519216

TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG

2021

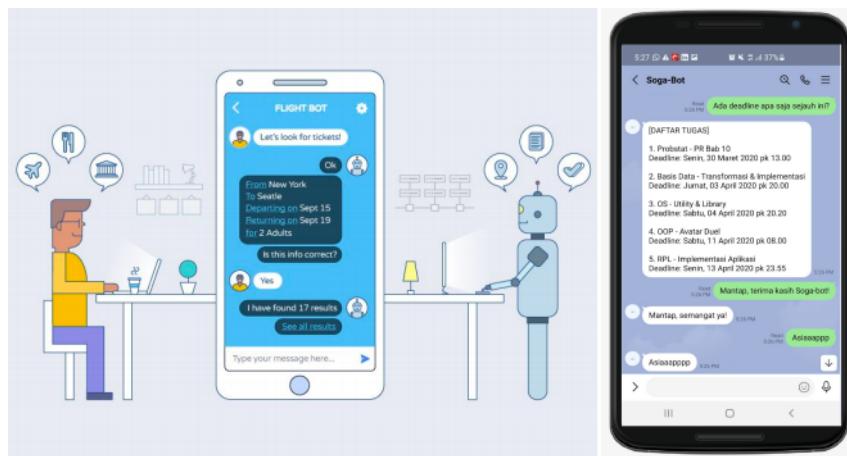
BAB 1 Deskripsi Tugas	3
1. Latar Belakang	3
2. Deskripsi Tugas	4
3. Fitur Aplikasi	4
4. Spesifikasi Program	9
5. Lain-Lain	10
6. Isi Laporan	11
7. Keterangan Laporan	12
8. Penilaian	12
BAB 2 Landasan Teori	13
1. Deskripsi Singkat Algoritma	13
a. Algoritma Boyer Moore	13
b. Regular Expression	15
2. Penjelasan Singkat ChatBot	16
BAB 3 Analisis Pemecahan Masalah	17
1. Langkah Penyelesaian Masalah	17
2. Fitur Fungsional dan Arsitektur Chatbot	18
BAB 4 Implementasi dan Pengujian	20
1. Spesifikasi Teknik Program	20
2. Tata Cara Penggunaan Program	24
3. Hasil pengujian	26
4. Analisis hasil pengujian	32
BAB 5 Kesimpulan	34
1. Kesimpulan	34
2. Komentar atau Refleksi	34

BAB 1

Deskripsi Tugas

1. Latar Belakang

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 melewatkkan deadline tugas.

2. 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.

3. Fitur Aplikasi

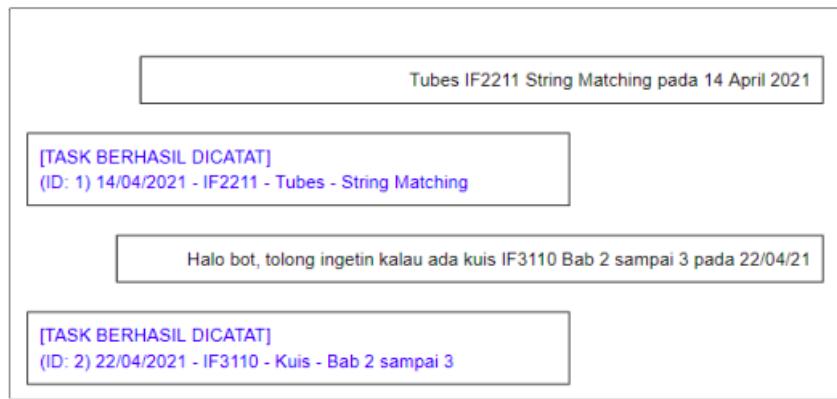
Deadline Reminder Assistant akan dibangun dengan sistem Question and Answer dimana pengembang diharapkan sudah menyediakan kumpulan formula tertentu untuk melakukan pendekripsi setiap perbedaan command atau perintah pada aplikasi Chatbot. Berikut ini adalah runtutan fitur yang dimiliki oleh Deadline 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

- 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?"

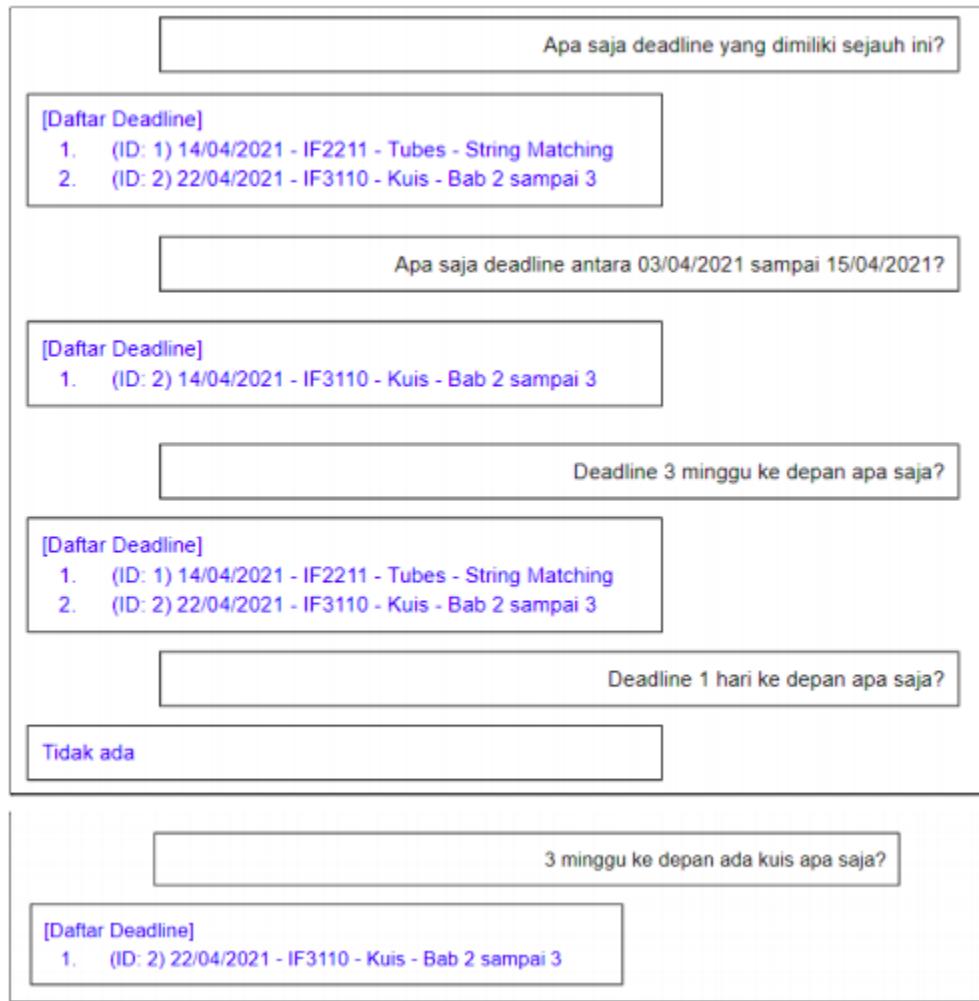
c. Berdasarkan jenis task (kata penting)

- 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

Catatan:

Eksekusi perintah pengguna bisa mencakup ketiga poin sekaligus sehingga formula pengenalan command sebaiknya dibuat sebagai satu kesatuan utuh.

Contoh interaksi



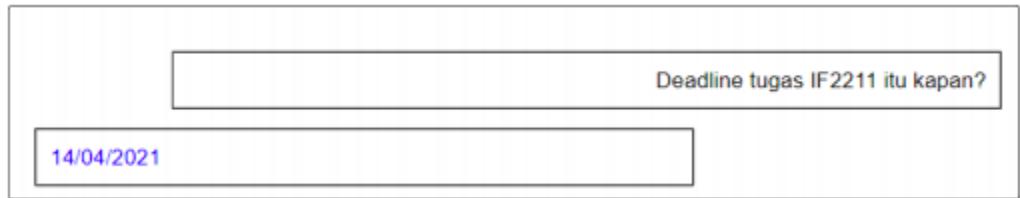
Keterangan penting:

- 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

- b. Misalnya: “Deadline tugas IF2211 itu kapan?”
- c. Contoh interaksi



4. Memperbarui 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 memperbarui 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 diperbarui, Chatbot akan menampilkan pesan sukses memperbarui 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

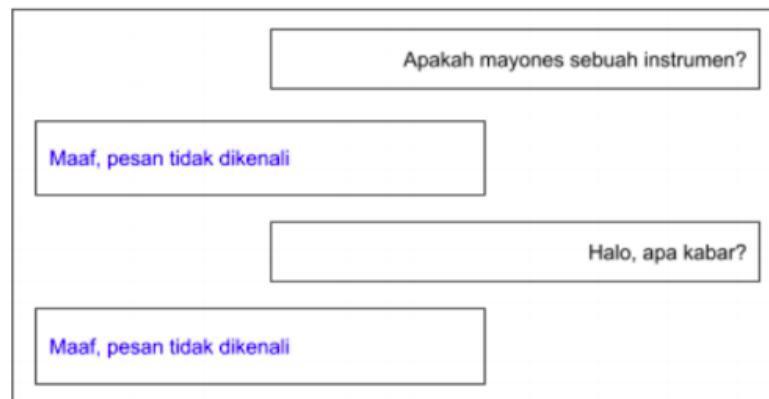


7. Mendefinisikan list kata penting terkait apakah itu merupakan suatu task atau tidak

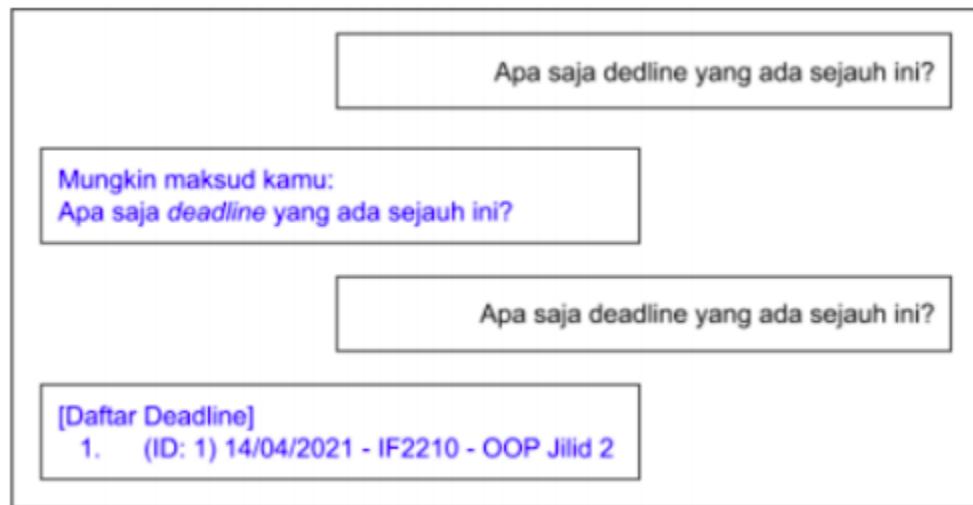
- a. Minimal terdapat 5 kata penting berbeda, contohnya adalah: [“Kuis”, “Ujian”, “Tucil”, “Tubes”, “Praktikum”]
- b. Kata penting akan digunakan pada penentuan jenis tugas dari suatu task.
- c. Daftar kata penting tidak perlu dibuat dinamis, cukup static saja atau hardcoded.

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
- Berikan rekomendasi kata jika perintah masukan pengguna mismatch dengan daftar kata yang diterima chatbot, namun masih memiliki tingkat kemiripan di atas 75%.
 - Contoh interaksi



- Ada berbagai metriks 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.

4. Spesifikasi Program

- Aplikasi yang dibuat berbasis web (wajib) dan anda dapat menggunakan salah satu kakis website: PHP, Flask, Django, JavaScript.
- Aplikasi (backend) harus menggunakan algoritma pencocokan string KMP, Boyer-Moore, dan Regex dengan menggunakan bahasa yang menunjang regular expression: Java, Javascript, PHP, Python.
- Penyimpanan data-data dan pengetahuan yang diperlukan oleh Chatbot bisa didefinisikan melalui 2 cara (pilih salah satu), yaitu:
 - Membuat suatu database sederhana (penerapan Basis Data dalam Strategi Algoritma). Implementasi skema database (relasi, atribut) dibebaskan. Skema basis data tidak perlu dinormalisasi.
 - 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 berikut.
 - a. List kata-kata penting
 - b. Daftar task yang tercatat oleh Chatbot
 - c. Data-data pendukung lainnya (kreativitas kelompok) IF2211 Strategi Algoritma - Tugas Besar 3 7
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.

5. Lain-Lain

1. Anda dapat menambahkan fitur-fitur lain yang menunjang program yang anda buat (unsur kreativitas).
2. Tugas dikerjakan berkelompok, minimal 2 orang dan maksimal 3 orang, boleh lintas kelas namun tidak boleh sekelompok dengan orang yang sama dengan tubes stima sebelumnya.
3. Semua kelompok harap mengisi data kelompok mereka pada link <http://tiny.cc/PendataanStima3>
4. Anda harus membuat aplikasi dan program ini sendiri kecuali library file dan regex, tetapi belajar dari contoh-contoh program serupa yang sudah ada tidak dilarang (tidak boleh melakukan plagiasi source code dari program orang lain). Program harus dibuat sendiri, tidak boleh sama dengan teman.
5. Program harus modular dan mengandung komentar yang jelas.
6. Beri nama tokoh di ChatBot ini, misalnya Sogabot, Hansbot, Awoobot, dll. Lengkapi dengan gambar avatarnya. Avatar yang dinamis (bibir bisa bergerak, gerakan tubuh) adalah kreativitas yang dihargai.
7. Dilarang menggunakan kode program yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, tetapi belajar dari program yang sudah ada tidak dilarang.
8. Batas akhir pengumpulan tugas adalah Rabu, 28 April 2021 pukul 23.59

- WIB. Keterlambatan dalam mengumpulkan akan diberi penalti pengurangan skor yang cukup signifikan.
9. Semua pertanyaan menyangkut tugas ini dapat dikomunikasikan lewat QnA yang bisa diakses pada <http://tiny.cc/QnATubes3>
 10. Bonus (maksimal 15 poin):
 - a. Mendeploy aplikasi web yang telah dibangun (hosting provider dibebaskan). Deployment website harus dipertahankan sampai demo tugas besar.
 - b. Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke Youtube. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video Youtube tersebut dan memutarnya IF2211 Strategi Algoritma - Tugas Besar 3 8 di depan asisten. Beberapa contoh video tubes tahun-tahun sebelumnya dapat dilihat di YouTube dengan menggunakan kata kunci "Tubes Stima", "Tugas besar stima", "strategi algoritma", dll.
 - c. Membuat fitur rekomendasi kata apabila terdapat kesalahan kata pada perintah.
 11. Demo akan dilakukan, tunggu informasi lanjut setelah waktu pengajaran tugas berakhir.
 12. Setiap anggota kelompok harus memahami seluruh program, termasuk bagian yang bukan bagian mereka.
 13. Program disimpan dalam folder Tubes3_NIM dengan NIM merupakan NIM anggota terkecil. Berikut merupakan struktur dari isi folder tersebut.
 - a. Folder src berisi source code
 - b. Folder doc berisi laporan tugas besar dengan format nama_kelompok.pdf
 - c. Folder test berisi data-data atau pengetahuan awal yang dimiliki oleh Chatbot. Apabila menggunakan database lokal, hasil dumpnya dapat disimpan disini.
 - d. README selengkap mungkin. Referensi README dapat diakses pada <https://github.com/ritaly/README-cheatsheet> atau referensi lain yang serupa.
 14. Folder tersebut di-zip dengan format yang sama dengan nama folder. Link pengumpulan akan diberitahukan lebih lanjut oleh asisten.

6. Isi Laporan

- Cover: Cover laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo "gajah" ganesha.

- Bab 1: Deskripsi tugas (dapat menyalin spesifikasi tugas ini).
- Bab 2: Landasan Teori.
 - Deskripsi singkat algoritma KMP, BM, dan Regex
 - Penjelasan singkat mengenai Chatbot
- Bab 3: Analisis Pemecahan Masalah.
 - Langkah penyelesaian masalah setiap fitur
 - Fitur fungsional dan arsitektur Chatbot yang dibangun
- Bab 4: Implementasi dan pengujian.
 - Spesifikasi teknis program (struktur data, fungsi, prosedur yang dibangun)
 - Penjelasan tata cara penggunaan program (interface program, fitur-fitur yang disediakan program, dan sebagainya)
 - Hasil pengujian (screenshot antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur dalam Chatbot)
 - Analisis hasil pengujian
- Bab 5: Kesimpulan, saran, dan komentar/refleksi tentang tugas besar 3 ini.
- Daftar Pustaka.

7. Keterangan Laporan

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

8. Penilaian

1. Bagian 1 : Laporan (25%)
 - a. Langkah penyelesaian masalah setiap fitur (10%)
 - b. Hasil pengujian dan analisis algoritma (10%)
 - c. Komponen-komponen lain dalam laporan (5%)
2. Bagian 2 : Implementasi Program (75%)
 - a. Kebenaran program (30%)
 - b. Pemahaman terhadap cara kerja program (25%)
 - c. Interface, Features, dan Unsur Kreativitas (20%)
3. Bagian 3 : Bonus (15%)
 - a. Melakukan deployment ke dalam website
 - b. Membuat video demonstrasi program
 - c. Membuat fitur rekomendasi kata apabila terjadi kesalahan

BAB 2

Landasan Teori

1. Deskripsi Singkat Algoritma

a. Algoritma Boyer Moore

Algoritma Boyer-Moore adalah salah satu algoritma pencarian string, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini disebut sebagai algoritma pencocokan string paling efektif pada saat ini. Algoritma ini berbeda dengan algoritma pencocokan string lainnya karena algoritma ini memulai pencocokan string mulai dari kanan yang kemudian mengarah ke kiri. Dengan begitu, akan lebih banyak informasi yang didapat.

Algoritma Boyer-Moore melakukan pencocokan string berdasarkan 2 teknik. Teknik tersebut adalah sebagai berikut.

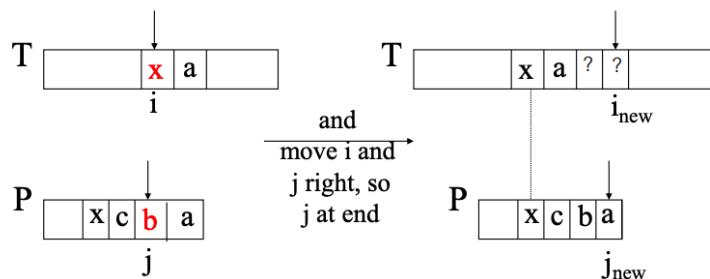
1. Teknik *looking-glass*

Teknik ini menekankan bahwa akan terdapat 2 string yaitu P dan T dimana P merupakan string yang akan dicari di dalam T. Kemudian, antara P dan T akan dicocokan dimulai dari huruf terakhir dari P bergerak mundur ke huruf pertama dari P.

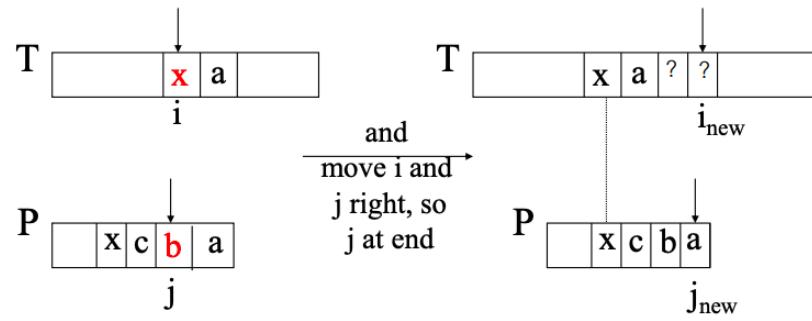
2. Teknik *character-jump*

Teknik ini menekankan bahwa ketika terjadi ketidakcocokan pada saat pencocokan string, maka akan dilakukan prosedur lompatan karakter. Terdapat 3 prinsip pelompatan:

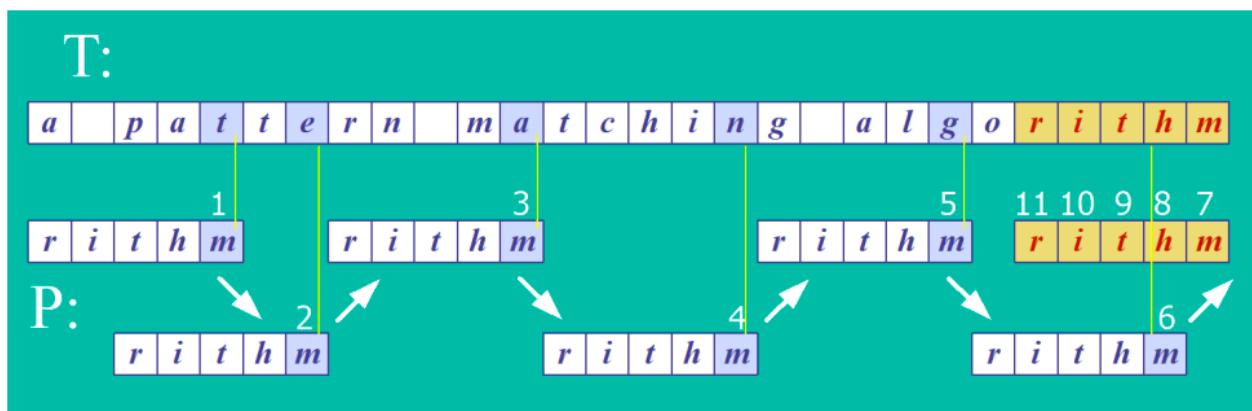
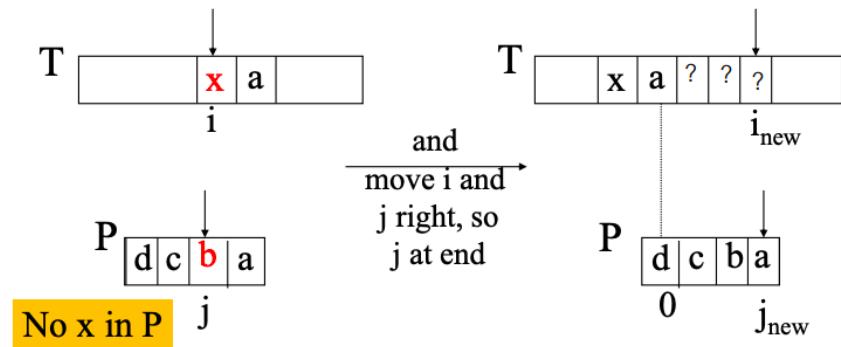
- Jika P mengandung x, kemudian shift right P sehingga antara x pada P dengan kemunculan terakhir x pada T sejajar.



- Jika shift right pada P tidak berhasil menemukan karakter yang sama dengan kemunculan terakhir x pada T, maka geser P ke kanan sehingga sejajar dengan T[i+1]



- c. Jika kasus a dan b tidak terpenuhi, maka geser P sehingga $P[0]$ sejajar dengan $T[i+1]$



b. Regular Expression

Regular Expression merupakan serangkaian karakter yang mendefinisikan suatu pola pencarian. Regular Expression membantu dalam menemukan string yang dimaksud pada suatu string masukan. Algoritma ini berbeda dengan exact matching yang mencari string yang tepat sesuai dengan string yang diinginkan. Dengan adanya regular expression, pengguna dapat melakukan pencarian pada string dengan lebih leluasa (misalnya mengabaikan karakter pemisah antara dua kata karena yang terpenting adalah kata tersebut muncul). Regular expression dikembangkan pada keilmuan komputer pada teori bahasa formal. Regular Expression digunakan dalam mesin pencari, dialog cari dan ganti dalam pengolah kata dan penyunting teks, dalam utilitas pengolahan teks dan dalam analisis leksikal. Kebanyakan bahasa pemrograman menyediakan pengolah ekspresi reguler baik secara bawaan atau melalui pustaka.

regular expressions	
Anchors	Sample Patterns
^ Start of line +	([A-Za-z0-9]+)
\A Start of string +	(\d{1,2})\vd{1,2}\vd{4})
\$ End of line +	((^\\$)+(?=\.jpg gif png)\\$)
\Z End of string +	(^\{1-9\}(1\\$ ^1-4\}(1\{0-9\}\{1\\$ \{50\\$)
\b Word boundary +	(#\{([A-Fa-f0-9])\{3}((A-Fa-f0-9)\{3}\)?
\B Not word boundary +	((?=.*\d)(?=.*[a-z])(?=.*[A-Z]).\{8,15\})
\< Start of word	(\w+@[a-zA-Z_]+\?,[a-zA-Z]\{2,6\})
\> End of word	(\</?[^>]+\>)
Character Classes	
\c Control character	Letters, numbers and hyphens
\s White space	Date (e.g. 21/3/2006)
\S Not white space	Jpg, gif or png image
\d Digit	Any number from 1 to 50 inclusive
\D Not digit	Valid hexadecimal colour code
\w Word	8 to 15 character string with at least one upper case letter, one lower case letter, and one digit (useful for passwords).
\W Not word	Email addresses
\xhh Hexadecimal character hh	HTML Tags
\Oxxx Octal character xxx	
POSIX Character Classes	
[:upper:] Upper case letters	Note These patterns are intended for reference purposes and have not been extensively tested. Please use with caution and test thoroughly before use.
[:lower:] Lower case letters	
[:alpha:] All letters	
[:alnum:] Digits and letters	
[:digit:] Digits	
[:xdigit:] Hexadecimal digits	
[:punct:] Punctuation	
[:blank:] Space and tab	
[:space:] Blank characters	
[:cntrl:] Control characters	
[:graph:] Printed characters	
[:print:] Printed characters and spaces	
[:word:] Digits, letters and underscore	
Assertions	
?= Lookahead assertion +	
?! Negative lookahead +	
?<= Lookbehind assertion +	
?!= or ?<! Negative lookbehind +	
?> Once-only Subexpression	
?() Condition [if then]	
?()! Condition [if then else]	
?# Comment	
String Replacement (Backreferences)	
\$n nth non-passive group	Note Ranges are inclusive.
\$2 "xyz" in /^(abc(xyz))\$/	
\$1 "xyz" in /^(?:abc)(xyz)\$/	
\$` Before matched string	
\$' After matched string	
\$+ Last matched string	
\$& Entire matched string	
\$__ Entire input string	
\$\$ Literal "\$"	
Metacharacters (must be escaped)	
^ [.	
\$ { *	
(\ +	
) ?	
< >	

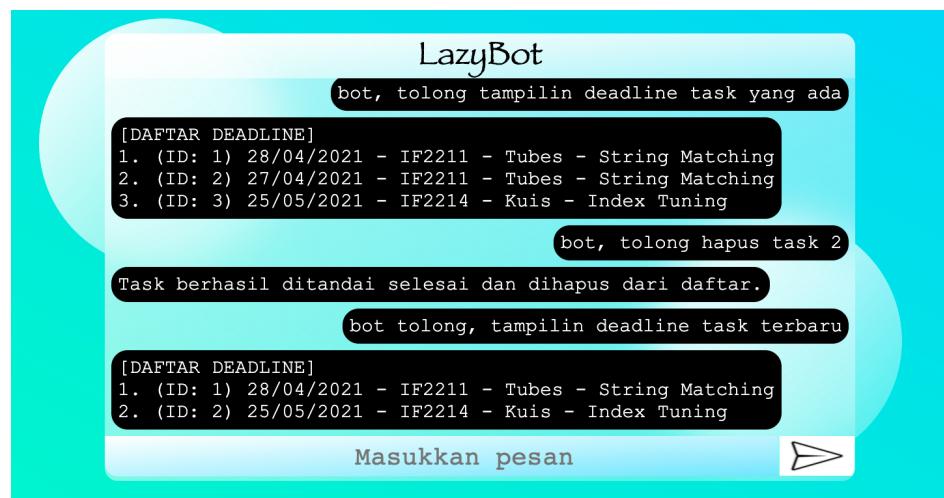
Available free from
AddedBytes.com

2. Penjelasan Singkat ChatBot

Chatbot yang kami buat bernama LazyBot yang dikembangkan berbasis *web application*. Chatbot ini berguna untuk menyimpan *todo list* sehingga membantu pengguna dalam mengetahui hal apa saja yang perlu dilakukan.



Aplikasi ini mirip seperti trello dan app produktivitas lainnya. Perbedaan dan sekaligus keunikan dari aplikasi ini adalah berbentuk chatbot sehingga dalam penambahan task, pembacaan deadline task, dll dilakukan dengan mengirimkan pesan layaknya berbicara normal kepada bot. Setelah itu, bot akan memberikan informasi yang pengguna inginkan jika perintah masukan tersedia.



BAB 3

Analisis Pemecahan Masalah

1. Langkah Penyelesaian Masalah

Tahapan dalam penyelesaian tugas besar ini dapat dikelompokkan menjadi:

1. Desain Modul Backend

Pada tahap ini, dilakukan desain dan pengujian independen terhadap:

- a. Algoritma Boyer-Moore dalam bahasa Python (untuk digunakan kemudian ketika dilakukan perbandingan string).
- b. Regular Expression (Regex) yang dikhususkan untuk mendukung fitur penambahan task. Regex ini berperan untuk mendeteksi 4 komponen penting dari fitur tambah task, yaitu Tipe (“Tubes”, “Tucil”, dan sejenisnya), Kode Mata Kuliah, Topik, dan Tanggal.

2. Desain Fitur Utama Chatbot

Pada tahap ini, dibentuk sebuah modul terintegrasi yang berisi fungsi-fungsi untuk menangani identifikasi fitur Chatbot berdasarkan teks atau string yang diberikan. Pada modul ini terdapat 6 buah fungsi:

- a. cek_tambah_task (memanfaatkan modul regex pada tahap 1)
- b. cek_task (memanfaatkan Boyer-Moore)
- c. cek_deadline (memanfaatkan Boyer-Moore dan regex)
- d. cek_perbarui (memanfaatkan Boyer-Moore dan regex)
- e. cek_selesai (memanfaatkan Boyer-Moore dan regex)
- f. cek_help (memanfaatkan regex)

3. Simulasi Chatbot Sederhana

Pada tahap ini dibentuk sebuah file dalam bahasa Python yang menyertakan modul kumpulan fungsi sebelumnya. File ini dibuat dengan tujuan sebagai media simulasi sederhana (dalam CLI) dan mempermudah pengujian terhadap fungsi yang sudah dibentuk jika terjadi perubahan ke depannya.

4. Desain Modul Frontend

Setelah semua komponen backend yang diperlukan sudah tersedia, tahap berikutnya adalah implementasi tampilan Chatbot pada web browser dengan memanfaatkan web framework berupa Python Flask. Pada tahap ini dibentuk banyak file (untuk menurunkan kompleksitas sebuah file) tambahan baru untuk mendukung tampilan Chatbot. Bentuk final Chatbot ini dapat diakses dengan mengeksekusi main.py dan kemudian membuka alamat lokal pada sebuah web browser.

5. Revisi dan Testing Lanjutan

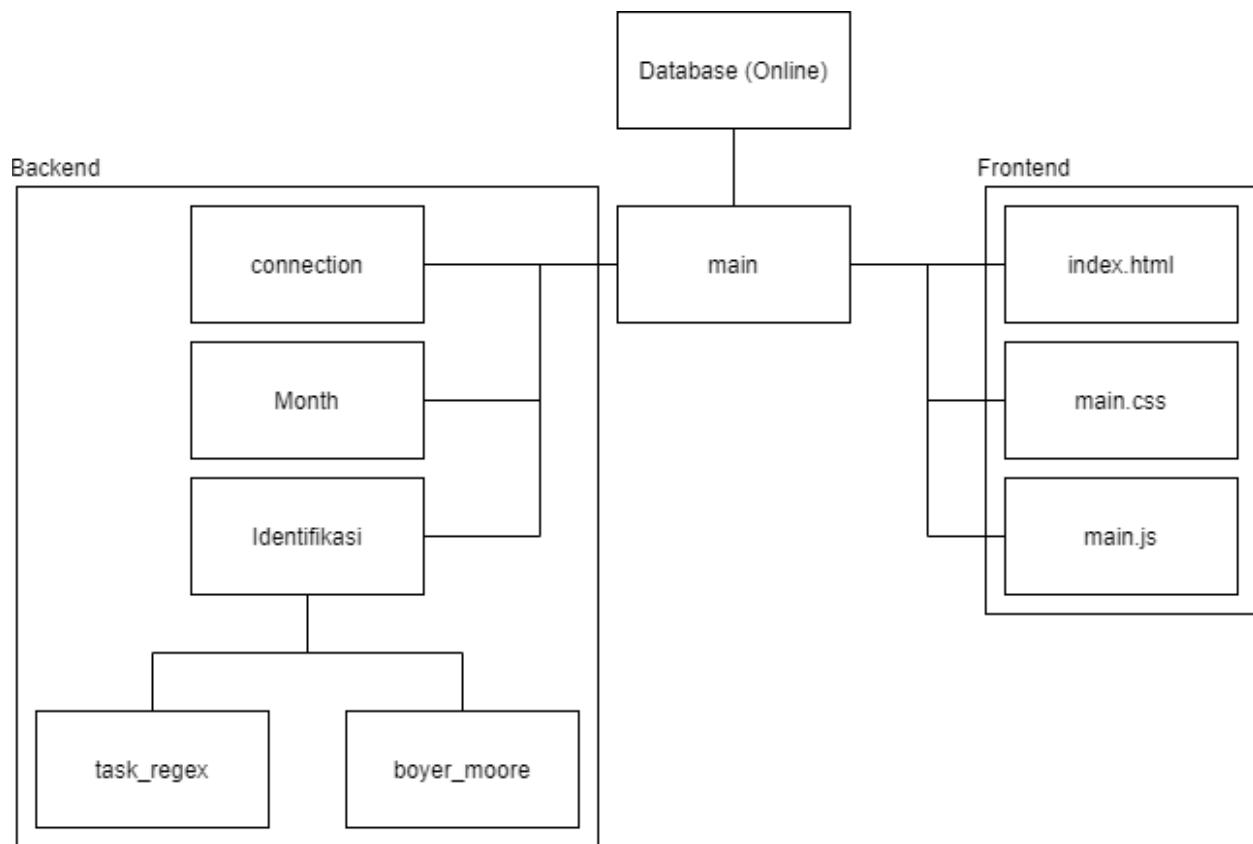
Tahapan tambahan ini sesuai namanya dilakukan untuk memperbaiki kesalahan-kesalahan yang muncul lalu menguji kode revisinya.

2. Fitur Fungsional dan Arsitektur Chatbot

Fitur yang diimplementasikan pada Chatbot sudah sesuai dengan spesifikasi utama, yaitu:

No.	Fitur	Keterangan
1.	Menambahkan task baru	Trigger: terdapat komponen Tipe, Kode Mata Kuliah, Topik, dan Tanggal (keseluruhan dituliskan dalam format terurut dan benar)
2.	Melihat daftar task	
2.a.	Seluruh task	Trigger: terdapat kata “deadline”
2.b.i	Periode antara dua tanggal	Trigger: terdapat kata “deadline” dan dua buah tanggal (dalam format benar)
2.b.ii	N minggu ke depan	Trigger: terdapat kata “deadline”, sebuah angka dan “minggu”
2.b.ii i	N hari ke depan	Trigger: terdapat kata “deadline”, sebuah angka, dan “hari”
2.b.i v	Hari ini	Trigger: terdapat kata “deadline”, dan “hari ini”
2.c	Berdasarkan jenis task (Tipe)	Trigger: terdapat kata petunjuk Tipe, sebuah angka, dan “minggu” / “hari”
3.	Menampilkan deadline suatu task	Trigger: terdapat kata “deadline” dan kode mata kuliah
4.	Memperbarui task tertentu	Trigger: terdapat kata “task”, sebuah angka, dan tanggal (dalam format benar)
5.	Menandai task tertentu sudah selesai	Trigger: terdapat kata “task” dan sebuah angka
6.	Menampilkan opsi help	Trigger: terdapat kata “help”, “bisa”, “petunjuk”
7.	Mendefinisikan list kata penting	Terdefinisi dalam file file_kata_tipe.txt
8.	Menampilkan pesan error	Trigger: perintah tidak teridentifikasi oleh semua fungsi identifikasi (pada 3.1.2)

Arsitektur Chatbot dapat dilihat pada diagram berikut:



BAB 4

Implementasi dan Pengujian

1. Spesifikasi Teknik Program

Index.html

Class / Id	Fungsi
container	Bagian jendela secara keseluruhan (a whole window). Tempat semua komponen menindih.
contents	Kotak yang berada di tengah jendela. Bagian ini layer setelah container. Bagian ini merupakan wadah untuk komponen heading, middle, dan bottom.
heading	Bagian ini merupakan header dan mewadahi tulisan "LazyBot".
middle	Wadah untuk penampilan pesan-pesan dari pengguna dan juga bot. Bagian ini dapat di-scroll sepanjang pesan yang ada.
bottom	Wadah untuk input text box dan submit button yang berada di bawah bagian middle.
masukan	Input text box. Bagian ini merupakan tempat pengguna memasukkan pesannya.
submit	Submit button. Bagian ini merupakan tombol bagi pengguna untuk mengirimkan pesan nya ke server.
pesan	Bagian ini mengandung pesan yang dikirim oleh user dan ditampilkan.
BOT	Bagian ini mengandung pesan balasan dari bot. Pesan dari bot ini dikirim setelah terdapat pesan dari pengguna.
circle1	Bagian dekorasi lingkaran yang berada pada kiri atas jendela. Bagian ini bukanlah fitur fungsional dari aplikasi.
circle2	Bagian dekorasi lingkaran yang berada pada kanan bawah jendela. Bagian ini bukanlah fitur fungsional dari program.

main.js

Fungsi	Tujuan
<code>\$("#form").on('submit',function(event)</code>	Prosedur yang mengatur aktivitas/aksi dari aplikasi ketika button submit ditekan.
<code>\$.ajax</code>	mengatur aplikasi agar ketika dilakukan request ke server dan menerima kembali response, halaman website tidak harus di reload.
<code>window.setInterval(function()</code>	Mengatur bagian tempat penampilan pesan (middle) agar secara otomatis menampilkan pesan terbaru dari pengguna.

main.py

Fungsi	Tujuan
<code>def index()</code>	Menjalankan/merender index.html.
<code>def process()</code>	Mengolah request dari pengguna. Setiap perintah dari pengguna akan diolah pada bagian ini. Di bagian ini pesan dari pengguna akan diklasifikasikan apakah merupakan perintah penambahan task, update data, menampilkan deadline , atau yang lainnya. Jika pesan tersebut tidak dikenal maka akan dikembalikan pesan “Pesanan tidak dikenal” melalui pesan BOT.
<code>def override_url_for()</code>	Berfungsi untuk mengatur ulang bentuk return value dari url_for (bukan bagian dari fitur fungsional program).
<code>def dated_url_for(endpoint, **values)</code>	Melakukan render otomatis terhadap index.html sehingga jika file index.html diubah tidak harus reconnect ke server (bukan bagian dari fitur fungsional program).

boyer_moore.py

Fungsi	Tujuan
def boyer_moore_search(text, pattern)	Fungsi ini menerapkan sistem pencocokan string boyer-moore. Fungsi ini mengembalikan True jika ditemukan dan False jika tidak.

Identifikasi.py

Fungsi	Tujuan
def cek_tambah_task(text)	Berfungsi untuk mengecek apakah text input mengandung perintah penambahan task. Fungsi ini akan mengembalikan true/false (berdasarkan apakah text input mengandung perintah penambahan task), mata kuliah yang ditambahkan, waktu dari task yang ditambahkan, tipe task yang ditambahkan, dan topik.
def cek_task(text)	Berfungsi untuk mengecek apakah string input mengandung perintah pengecekan task. Return value dari fungsi ini disesuaikan berdasarkan case masing-masing.
def cek_deadline(text)	Berfungsi untuk mengecek apakah string input mengandung perintah pengecekan deadline dari suatu mata kuliah. Return value dari fungsi ini adalah true/false (apakah text mengandung kata kunci) dan mata kuliah yang terdapat dalam input.
def cek_perbarui(text)	Berfungsi untuk mengecek apakah string input mengandung perintah untuk memperbarui waktu deadline dari task. Return value dari fungsi ini adalah true/false (apakah text mengandung kata kunci), id task yang ingin diganti, dan waktu terbaru.
def cek_selesai(text)	Berfungsi untuk mengecek apakah string input mengandung perintah untuk penandaan bahwa task telah selesai. Return value dari fungsi ini adalah true/false (apakah text mengandung kata kunci) dan id task yang ingin ditandai selesai.

<code>def cek_help(text)</code>	Berfungsi untuk mengecek apakah string input mengandung perintah help/bantuan. Return value dari fungsi ini adalah true/false (apakah text input mengandung kata kunci help/bisa/petunjuk).
---------------------------------	---

task_regex.py

Fungsi	Tujuan
<code>def regex_search(text)</code>	Berfungsi untuk mengecek apakah text input mengandung perintah penambahan task. Fungsi ini akan mengembalikan true/false (berdasarkan apakah text input mengandung perintah penambahan task), mata kuliah yang ditambahkan, waktu dari task yang ditambahkan, tipe task yang ditambahkan, dan topik.
<code>def regex_print(text)</code>	Berfungsi untuk menampilkan

connection.py

Fungsi	Tujuan
<code>class Connection</code>	Berfungsi sebagai tempat penyimpanan koneksi antara aplikasi dengan database.
<code>def __init__(self)</code>	Berfungsi untuk menginisialisasi koneksi dengan database. Koneksi tersebut akan disimpan ke dalam satu atribut kelas yaitu <code>db</code> . Koneksi akan diinisialisasi ketika kelas Connection telah diinstansiasi.

Month.py

Fungsi	Tujuan
<code>def convertMonth(month)</code>	Berfungsi untuk mengubah bulan dalam bentuk kata menjadi angka.
<code>def formattingTime(waktu)</code>	Berfungsi untuk mengubah format dari waktu input menjadi format <code>dd/mm/yyyy</code> .

2. Tata Cara Penggunaan Program

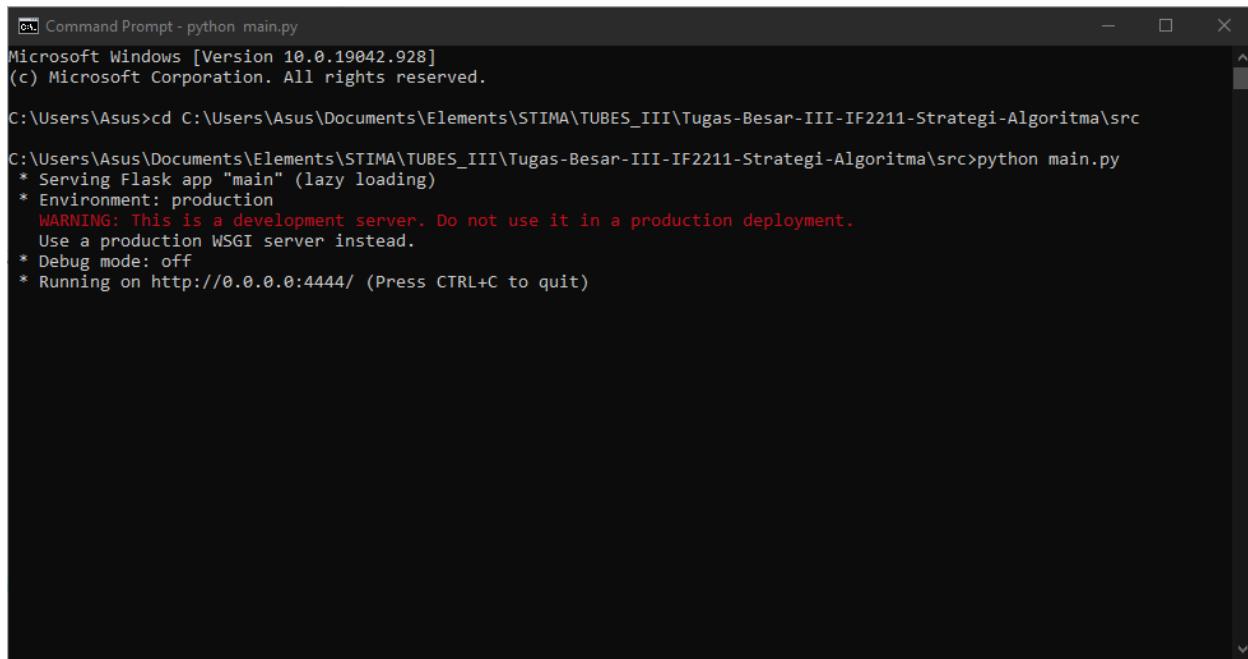
Untuk menjalankan program dengan baik dan benar, ada beberapa hal yang perlu dipersiapkan dan dipastikan, yaitu sebagai berikut:

- Terdapat python versi 3.9.1
- Menggunakan web browser (browser yang sudah teruji: Google Chrome dan Mozilla Firefox)
- Koneksi internet (karena menggunakan database untuk menyimpan daftar task)
- Folder src lengkap

Setelah semua persiapan lengkap, dapat dilakukan instruksi berikut:

1. Buka folder src yang tersedia, kemudian copy alamat direktori yang sedang dibuka.
2. Buka command prompt (cmd), kemudian masukkan perintah cd dan paste alamat tadi.
3. Setelah alamat working directory berubah, masukkan perintah “python main.py” atau “python3 main.py”.

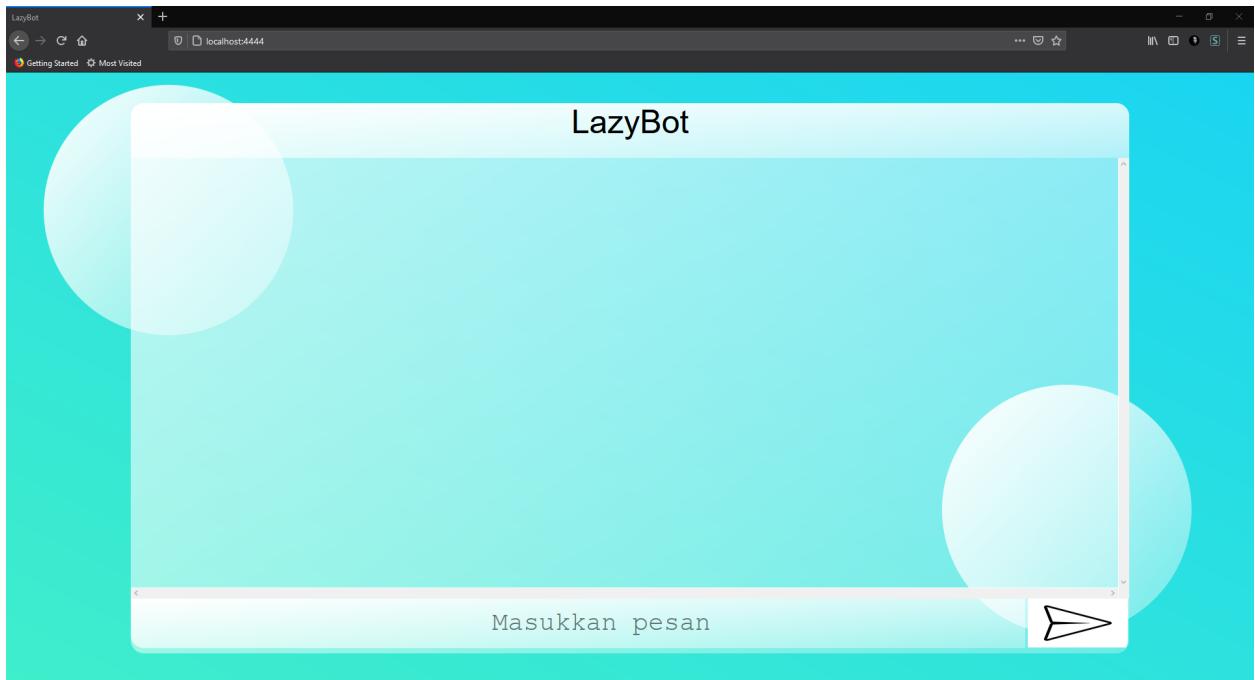
Layar cmd seharusnya berisi seperti berikut:



```
Command Prompt - python main.py
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>cd C:\Users\Asus\Documents\Elements\STIMA\TUBES_III\Tugas-Besar-III-IF2211-Strategi-Algoritma\src
C:\Users\Asus\Documents\Elements\STIMA\TUBES_III\Tugas-Besar-III-IF2211-Strategi-Algoritma\src>python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:4444/ (Press CTRL+C to quit)
```

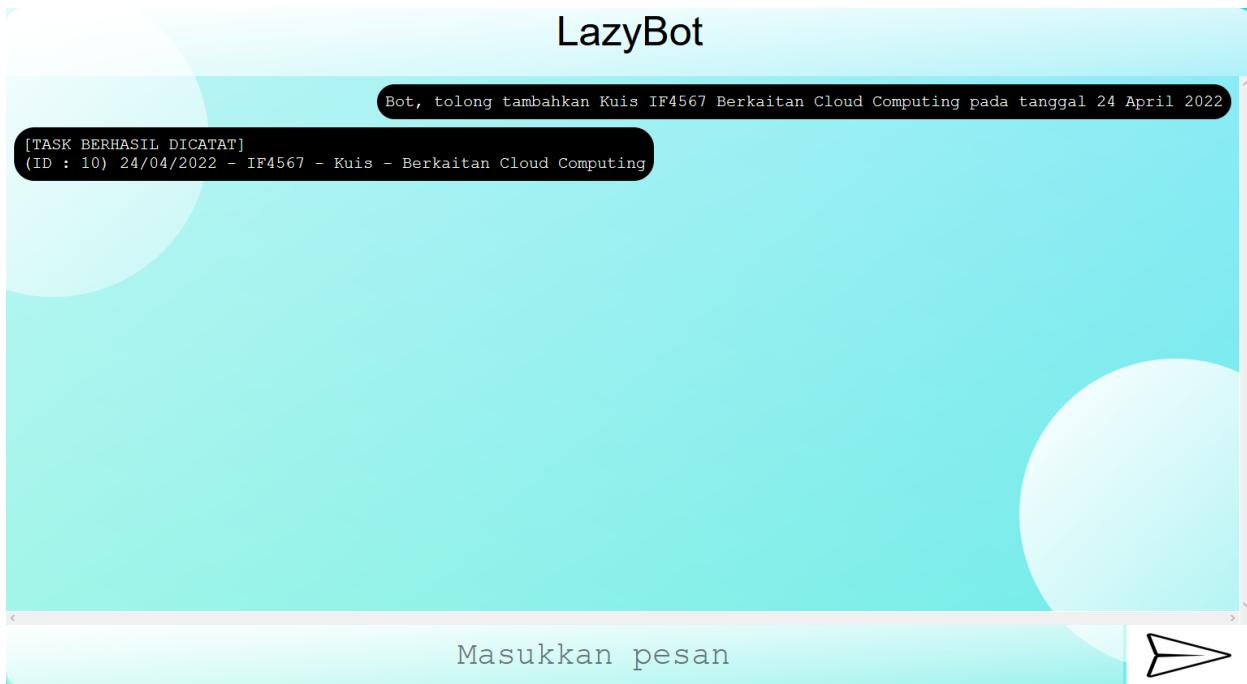
4. Abaikan Warning dengan tulisan merah tersebut, kemudian buka web browser yang akan digunakan.
5. Masukkan alamat "<http://localhost:4444/>"
6. Jika akses berhasil, pada web browser akan tampil seperti berikut:



7. Pada area berisi “Masukkan pesan”, dapat diisikan dengan teks perintah yang diinginkan
8. Setelah diberikan teks, dapat menekan tombol Enter atau mengklik ikon panah di sebelah kanan.
9. Terkait format perintah-perintah yang dapat diterima oleh chatbot dapat dilihat pada tabel di bagian 3.2.
10. Setelah bot selesai digunakan, web browser dapat dimatikan, dan pada cmd tadi lebih baik untuk diberi input CTRL + C untuk memberhentikan operasi.

3. Hasil pengujian

Menambahkan task baru:



Melihat daftar task:

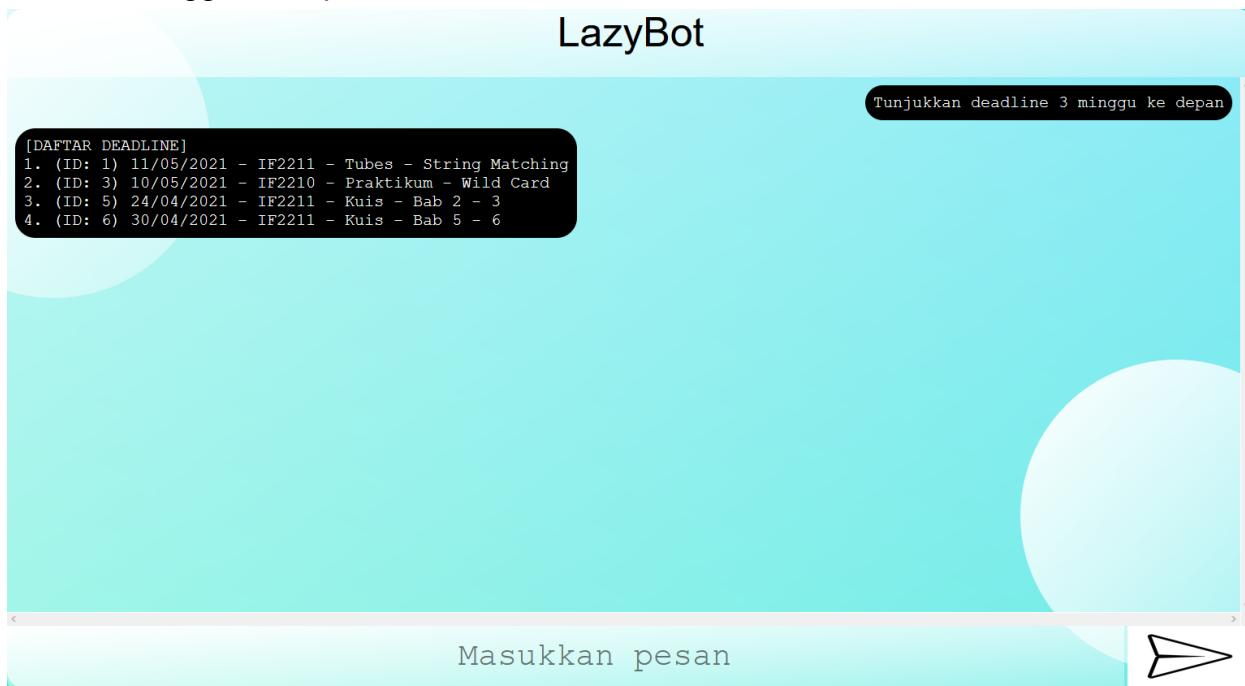
- Seluruh task:



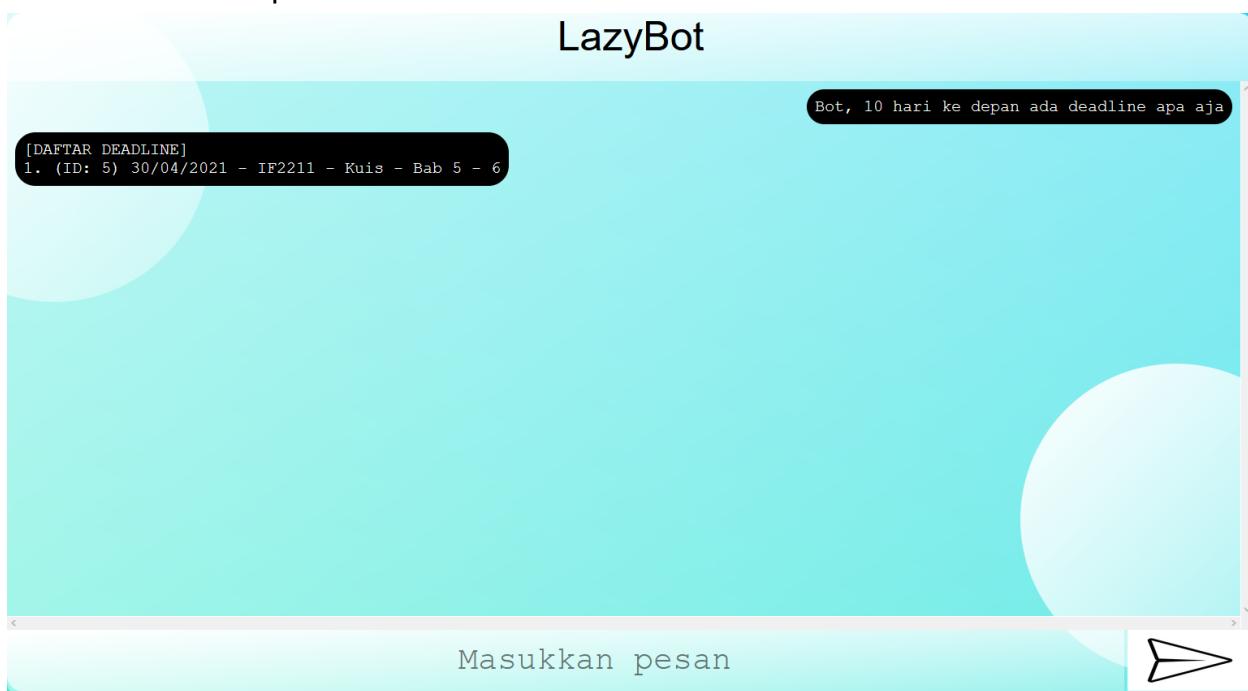
- Periode antara dua tanggal:



- N minggu ke depan:



- N hari ke depan:



- Hari ini:



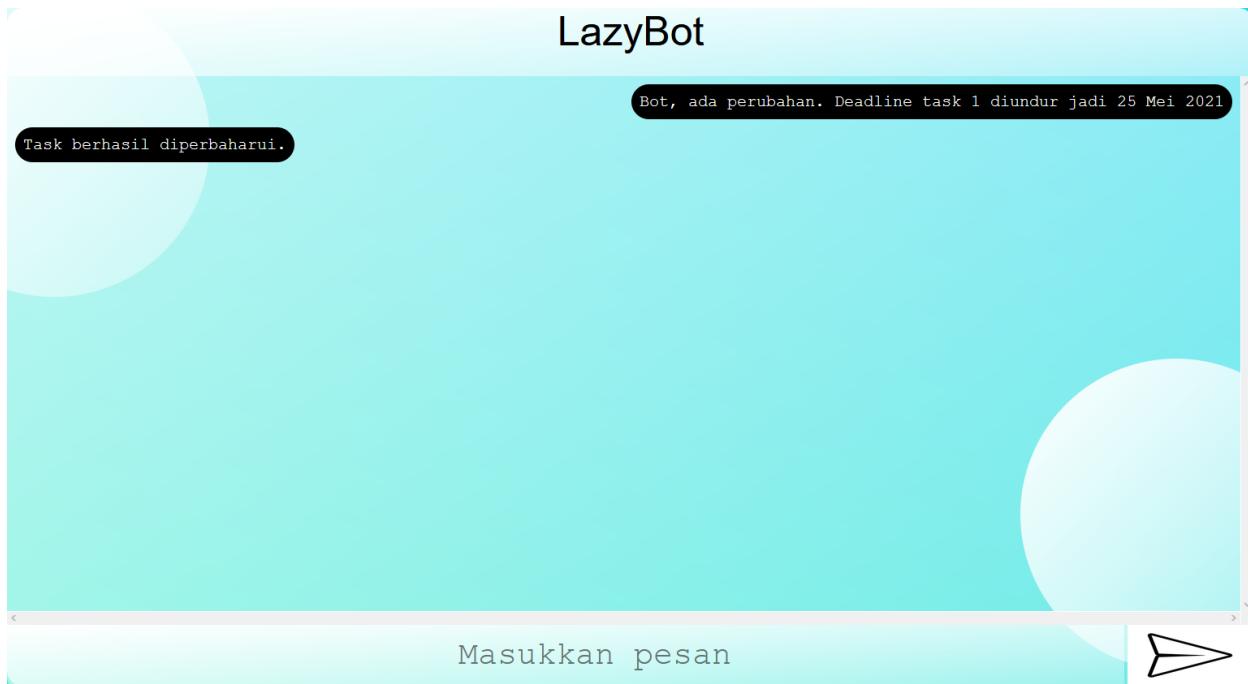
- Berdasarkan jenis task (Tipe):



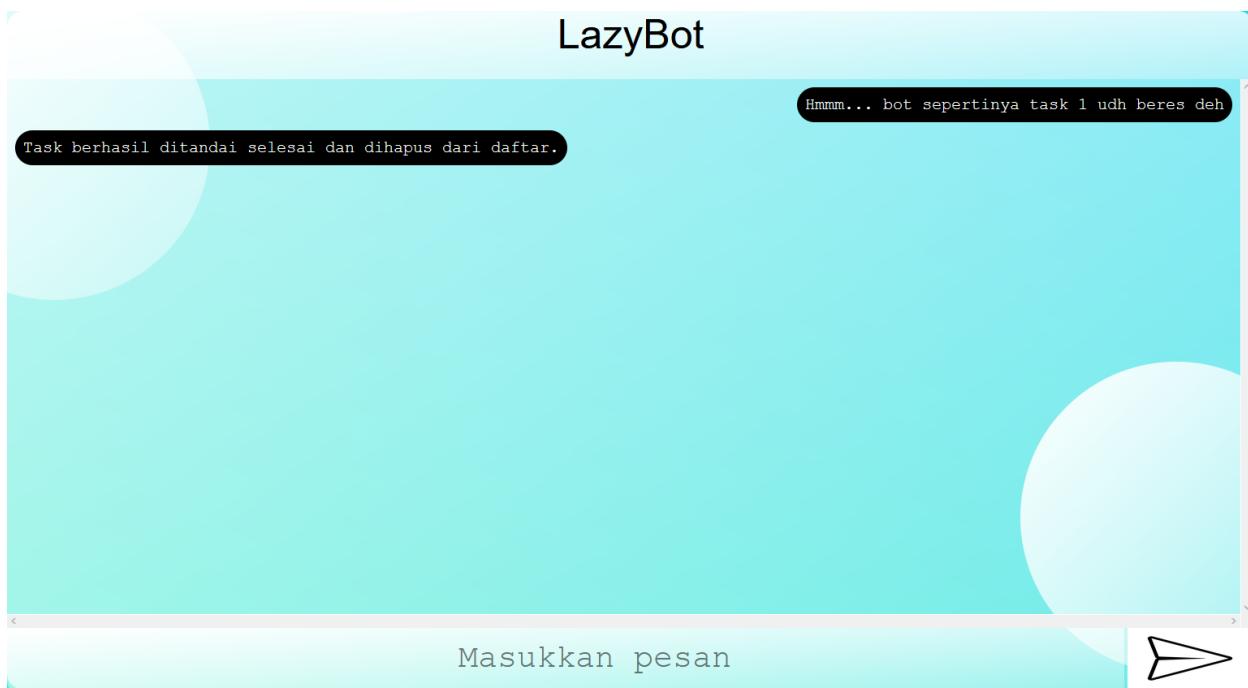
Menampilkan deadline suatu task:



Memperbarui task tertentu:

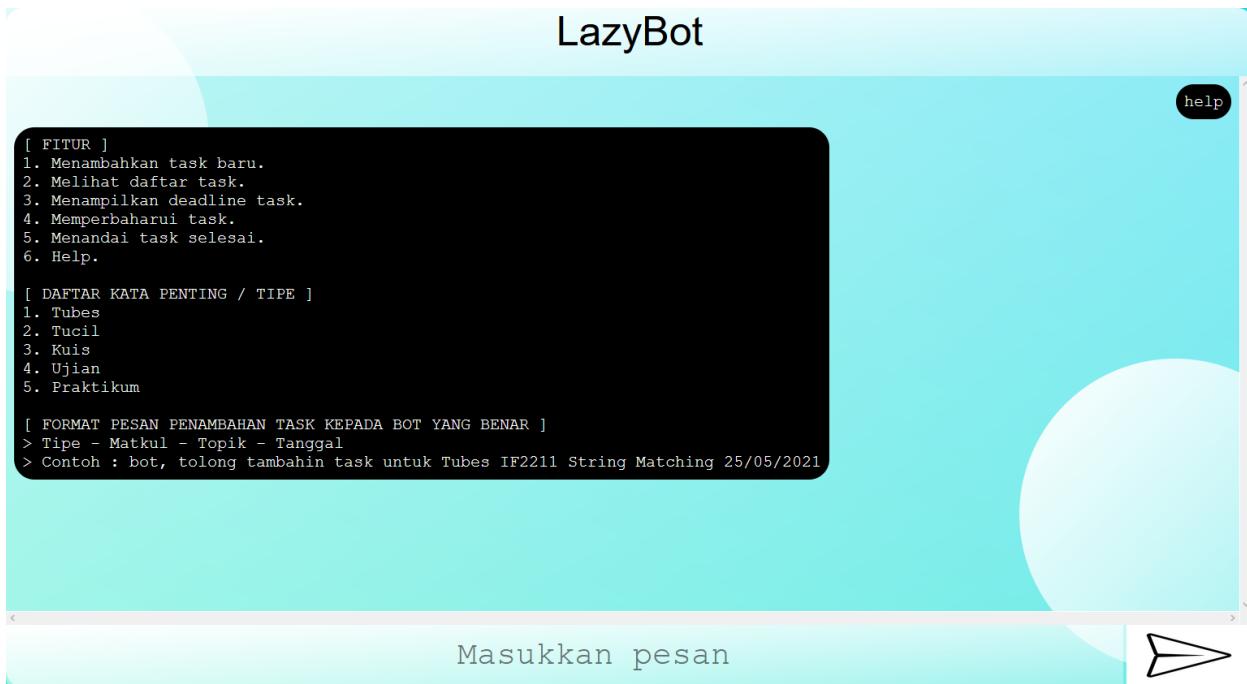


Menandai task tertentu sudah selesai:



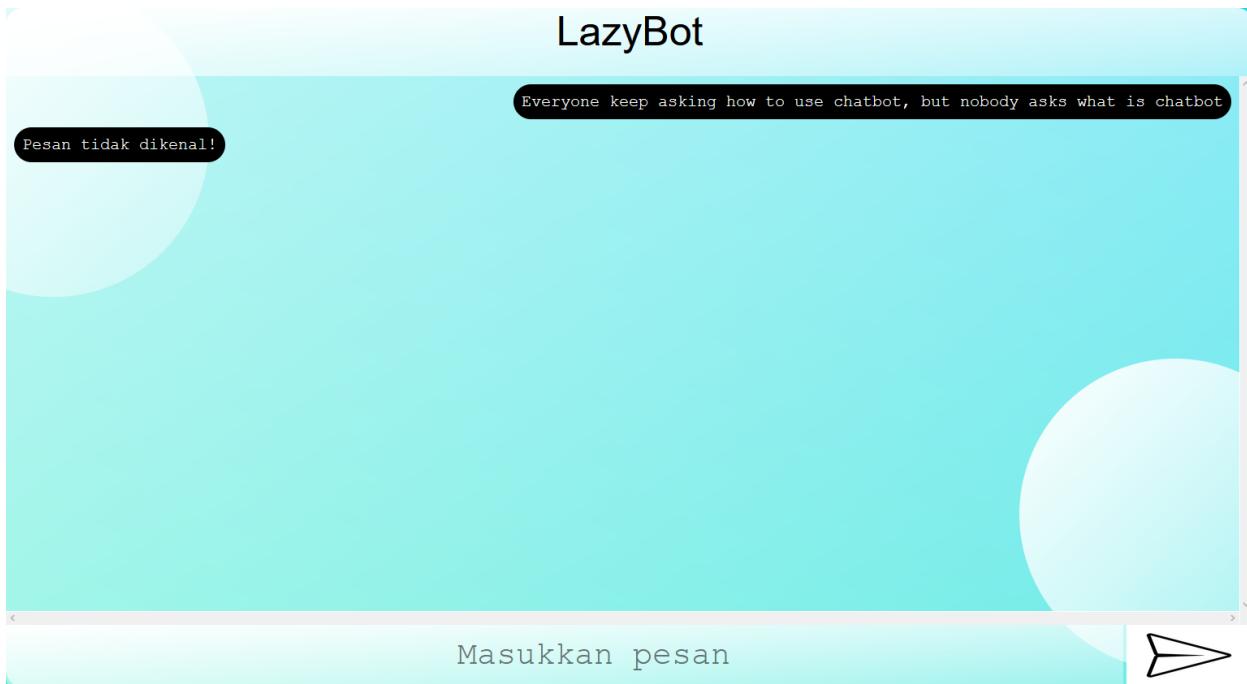
Menampilkan opsi help + Mendefinisikan list kata penting:

LazyBot



Menampilkan pesan error:

LazyBot



4. Analisis hasil pengujian

Dari hasil tangkapan layar pada bagian sebelumnya, ditunjukkan bahwa sudah semua fitur utama yang diminta dalam spesifikasi sudah diimplementasikan dan berhasil dijalankan atau diidentifikasi oleh chatbot yang dibuat. Namun, dikarenakan keterbatasan tempat untuk menampilkan seluruh hasil uji, pada hasil analisis ini langsung disampaikan saja masih adanya keterbatasan atau kekurangan dari chatbot yang dibuat ini.

Kekurangan dari chatbot yang ini terletak dari segi identifikasi perintah atau proses penerjemahan perintah dari bahasa natural menjadi instruksi yang hanya dimengerti program komputer. Contoh yang paling terlihat adalah pada gambar yang menampilkan opsi help. Di situ disertakan panduan format penulisan perintah yang benar agar dapat diidentifikasi oleh chatbot. Dengan demikian, user memiliki keterbatasan dalam memberikan masukan perintah.

Misalnya:

Format benar:

- Ada kuis IF2211 bab 2 - 3 pada tanggal 27 April 2022.

Format salah, namun dalam struktur bahasa natural dinilai benar:

- Pada 27 April 2022 ada kuis IF2211 yang materinya bab 2 - 3.
- Kuis IF2211 diadakan pada tanggal 27 April 2022 dengan materi bab 2 - 3.
- IF2211 akan mengadakan kuis pada tanggal 27 April 2022 dengan materi bab 2 - 3.
- IF2211 akan mengadakan kuis dengan bahan bab 2 - 3 pada 27 April 2022.
- Bab 2 - 3 akan dikuiskan pada matkul IF2211 tanggal 27 April 2022.

Kondisi bahasa natural yang sangat fleksibel menjadi alasan desain chatbot yang kaku dengan format demikian agar setidaknya dapat menerima perintah dan melakukan identifikasi dengan benar. Jika saja disediakan bentuk identifikasi dengan format lain untuk menangani variasi seperti di atas, maka secara matematis diperlukan $4!$ format (4 komponen yang dimaksud adalah Tipe, Kode Matkul, Topik, dan Tanggal).

Argumen lain yang menjadi alasan mengapa digunakan format perintah adalah untuk mempermudah pemenggalan atau klasifikasi setiap komponen, terutama komponen Topik yang terdiri atas banyak kata. Walaupun sudah digunakan regular expression yang tidak lagi peduli dengan letak kemunculan atau terjadinya perulangan, tetap saja tanpa format hasilnya tidak selalu sesuai dengan yang diharapkan.

Misalnya:

Dengan format, komponen Topik pada input “Ada kuis IF2211 bab 2 - 3 pada tanggal 27 April 2022” dapat diklasifikasikan sebagai rangkaian kata-kata setelah komponen Kode Mata Kuliah dan sebelum Tanggal.

Tanpa format:

- Pada 27 April 2022 ada kuis IF2211 yang materinya bab 2 - 3.

Topik adalah komponen di akhir kalimat, lebih spesifiknya setelah kata “materinya”.

- IF2211 akan mengadakan kuis dengan bahan bab 2 - 3 pada 27 April 2022.

Topik adalah komponen di antara Tipe dan Tanggal, lebih spesifiknya setelah kata “bahan”.

- Bab 2 - 3 akan dikuiskan pada matkul IF2211 tanggal 27 April 2022.

Topik adalah komponen di awal kalimat.

Karena dinilai tidak feasible untuk mencoba “meramal” seluruh kemungkinan yang input dari user, maka terpaksa dibuat format yang membatasi identifikasi chatbot tersebut, sehingga setidaknya chatbot ini dapat melakukan identifikasi perintah (terutama tambah task) dengan benar.

BAB 5

Kesimpulan

1. Kesimpulan

Berdasarkan percobaan yang dilakukan pada Bab 4, dapat ditarik kesimpulan bahwa berhasil dibuat sebuah chatbot dengan fitur-fitur yang tertera pada spesifikasi utama. Walaupun demikian, masih terdapat kekurangan pada chatbot ini terkait fleksibilitas dalam menerima input dari pengguna. Beberapa perintah memiliki format khusus agar teridentifikasi dengan tepat oleh chatbot, sehingga dengan kata lain tidak dapat sepenuhnya menerima perintah dalam bahasa natural.

2. Komentar atau Refleksi

Muhammad Rifky Muthahhari - 13519016

Sebenarnya agak capek sama tugas-tugas yang banyak banget dan gak abis-abis. Walaupun jadi tugas besar terakhir tapi lumayan buat jadi kerjaan. Mungkin lain kali tugasnya bisa disatuin gitu biar gak kayak banyak bat gini wkwkwk. Tapi isokeylah buat belajar dan nyoba bikin algoritmanya sendiri.

Aurelius Marcel Candra - 13519198

Sama seperti tubes-tubes sebelumnya, tubes kali ini juga memberikan sesuatu yang baru (walaupun tidak se-Wah tubes sebelum-sebelumnya). Setidaknya melalui tubes ini saya menjadi lebih mengerti maksud “^(.*)..*” (ketika melihat tutorial atau stackoverflow) dan bukanlah sebuah bahasa tidak dikenal atau kode rahasia. Tambahan, sejurnya saya tidak lagi berusaha begadang untuk menyelesaikan tuntas semua kombinasi tubes yang diberikan, entah karena ini dirasa mudah atau sudah tidak semangat atau tidak peduli lagi.

In a nutshell:



Kevin Katsura Dani Sitanggang - 13519216

Tugasnya itu not(not(not(not("seru"))))? seru : engga seru. Tugas STIMA memang selalu nambah pengalaman baru, new challenge and my new weapon comes. Tugas kali ini seru sih, apalagi berbau website beuhh seru bet.