

LAPORAN TUGAS BESAR III

IF2211 STRATEGI ALGORITMA

*Penerapan String Matching dan Regular Expression dalam Pembangunan Deadline
Reminder Assistant*



Disusun oleh
Kelompok 20 - AntiKeosKeosBot

Rexy Gamaliel Rumahorbo	13519010
Ruhiyah Faradishi Widiaputri	13519034
Sharon Bernadetha Marbun	13519092

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2021

DAFTAR ISI

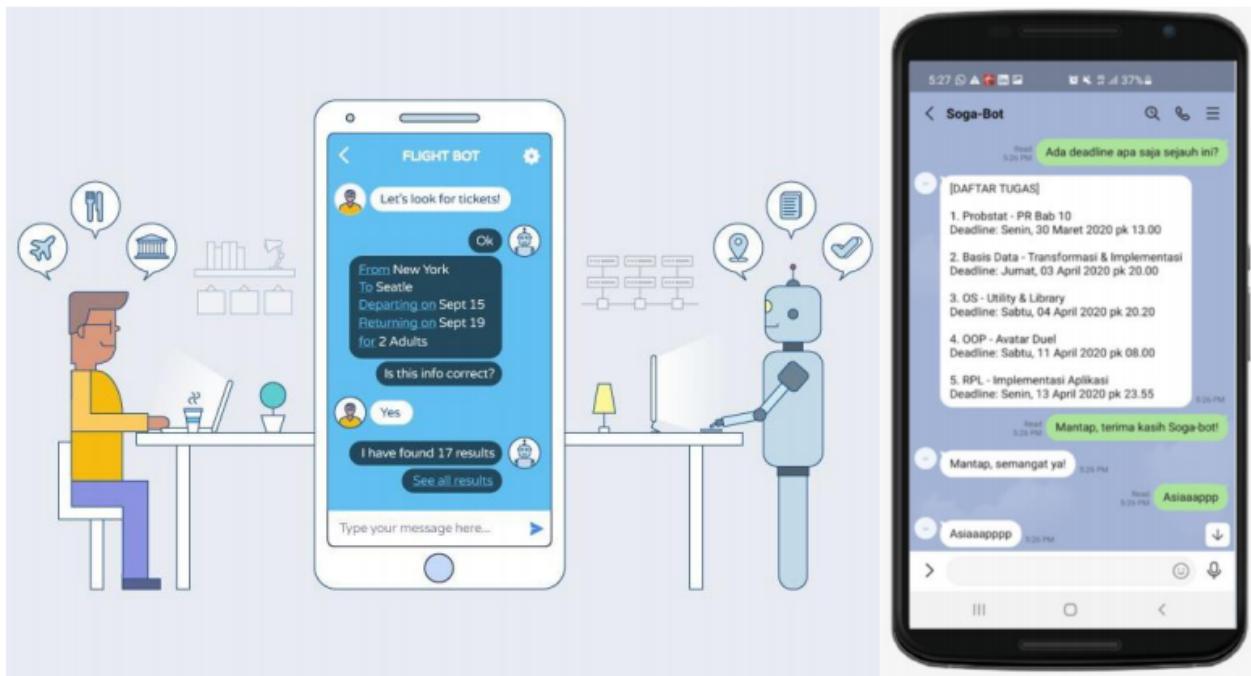
BAB I Deskripsi Tugas	2
Latar Belakang	2
Deskripsi Tugas	3
Fitur-Fitur Aplikasi	3
Spesifikasi Program	8
BAB II Landasan Teori	10
Algoritma Knuth-Morris-Pratt (KMP)	10
Algoritma Boyer Moore	10
Regular Expression (Regex)	11
Chatbot	12
BAB III Analisis Pemecahan Masalah	13
Langkah-Langkah Pemecahan Masalah	13
Fitur Fungsional dan Arsitektur Chatbot	15
BAB IV Implementasi dan Pengujian	19
Implementasi Program	19
Tata Cara Penggunaan Program	23
Hasil Pengujian	24
Analisis Hasil Pengujian	35
BAB V Kesimpulan dan Saran	37
Kesimpulan	37
Saran	37
Refleksi dan Komentar	37
DAFTAR PUSTAKA	38

BAB I

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*.



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

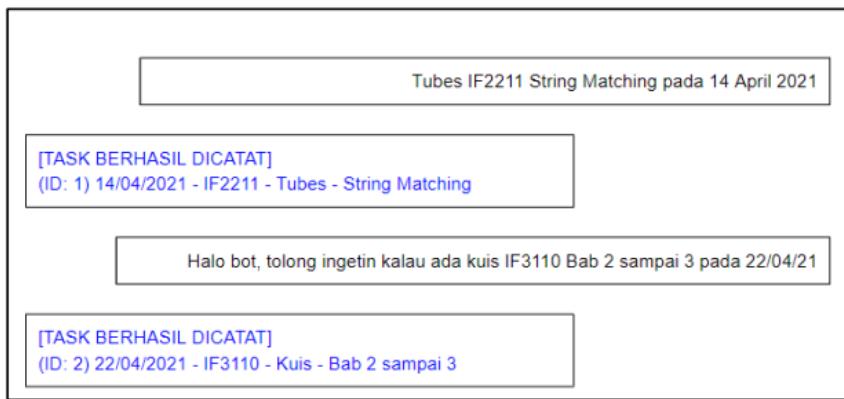
Dalam tugas besar ini, Anda akan diminta untuk membangun sebuah chatbot sederhana yang berfungsi untuk membantu mengingat berbagai deadline, tanggal penting, dan *task-task* tertentu kepada *user* yang menggunakannya. Dengan memanfaatkan algoritma *String Matching* dan *Regular Expression*, Anda dapat membangun sebuah chatbot interaktif sederhana layaknya Google Assistant yang akan menjawab segala pertanyaan Anda terkait informasi deadline tugas - tugas yang ada.

3. Fitur-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:
 - 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 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.
- 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

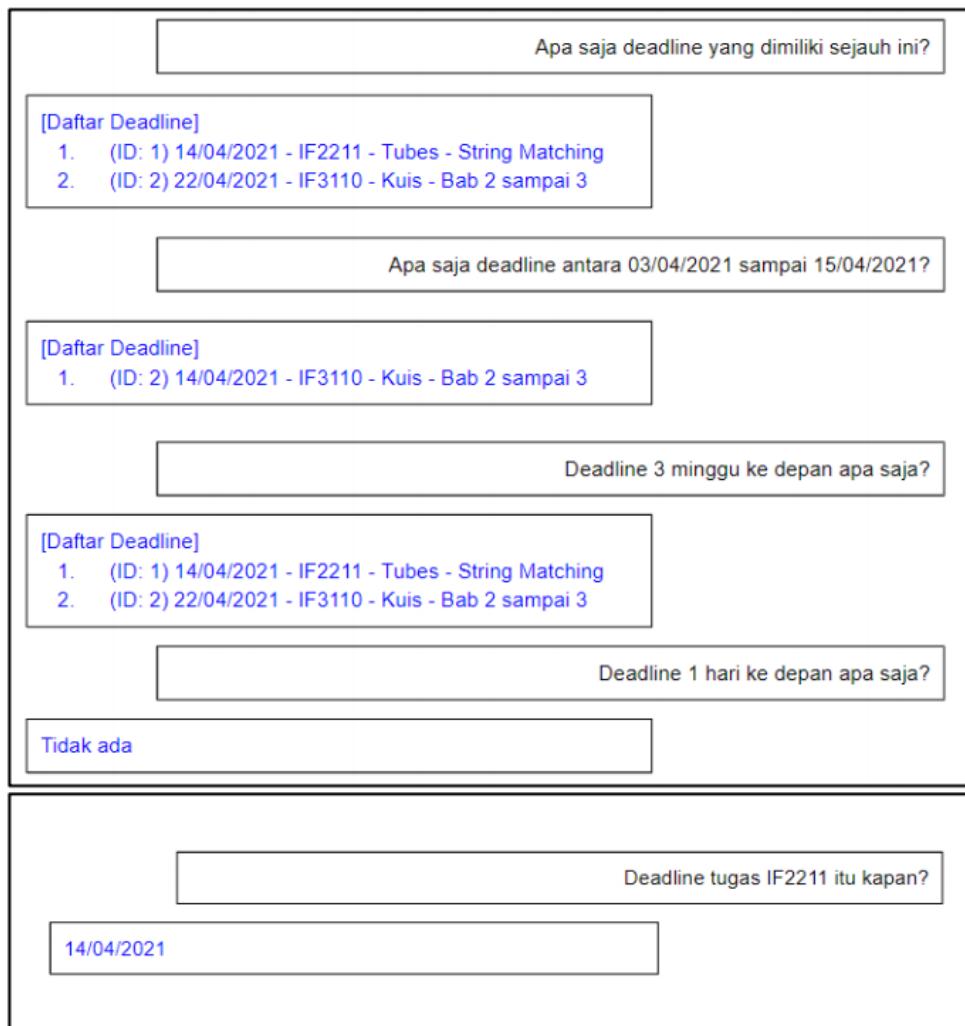
- Pada periode tertentu (DATE_1 until DATE_2) Contoh perintah yang dapat digunakan: “Apa saja deadline antara DATE_1 sampai DATE_2?”
- N minggu ke depan Contoh perintah yang dapat digunakan: “Deadline N minggu ke depan apa saja?”
- N hari ke depan Contoh perintah yang dapat digunakan: “Deadline N hari 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
- Misalnya: “Deadline tugas IF2211 itu kapan?”
- Contoh interaksi

	Deadline tugas IF2211 itu kapan?
14/04/2021	

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)

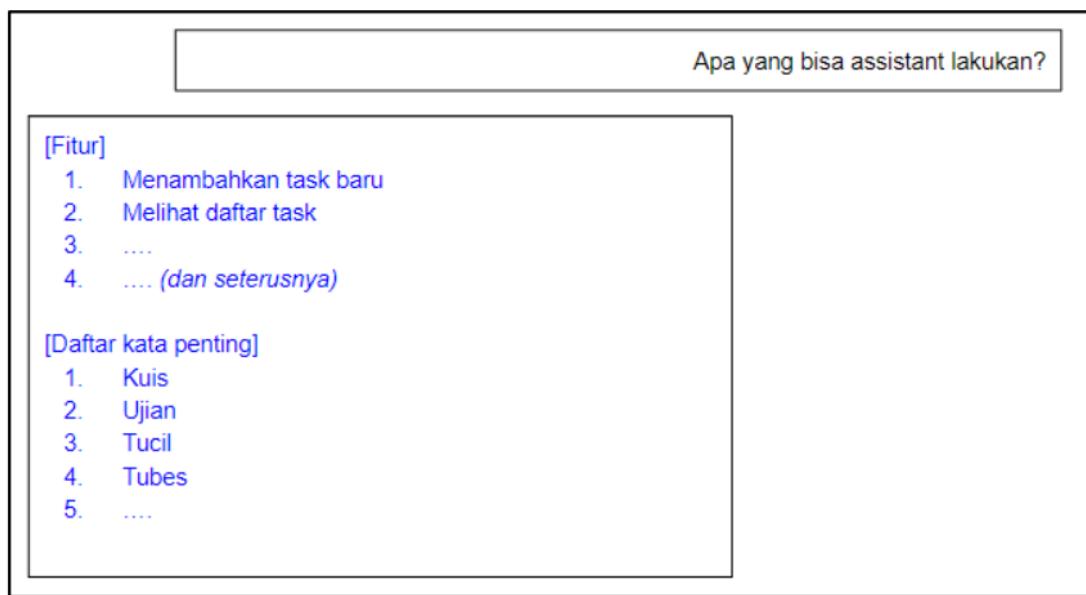
5. Menandai bahwa suatu task sudah selesai dikerjakan

- 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)

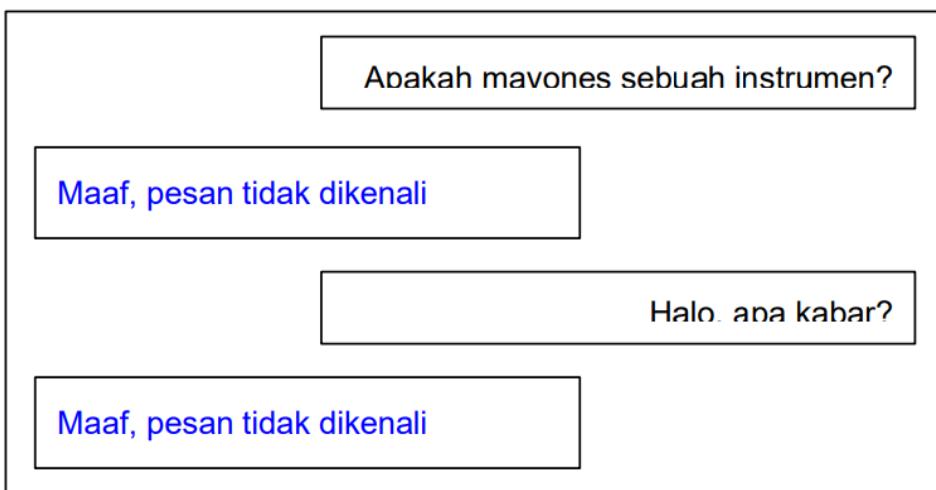
6. Menampilkan opsi *help* yang difasilitasi oleh *assistant*

- 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)

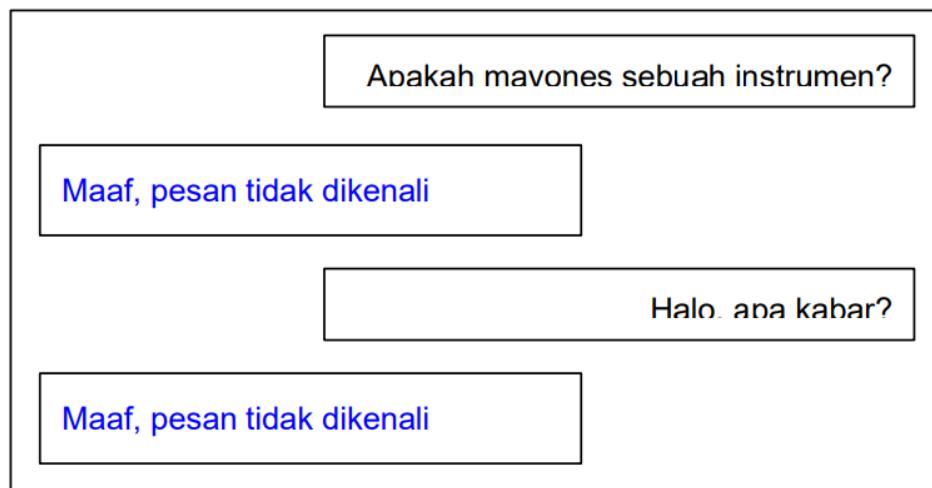
d. Contoh interaksi



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 [Understanding the Levenshtein Distance Equation for Beginners | by Ethan Nam | Medium.](#)

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.

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

BAB II

Landasan Teori

1. Algoritma Knuth-Morris-Pratt (KMP)

Algoritma KMP merupakan algoritma pencocokan string yang lebih “cerdas” daripada algoritma brute force. Dalam melakukan pencocokan string, pendekatan algoritma KMP mirip dengan algoritma brute force yaitu keduanya sama-sama melakukan pemeriksaan dari kiri ke kanan. Namun pergeseran yang dilakukan dalam algoritma KMP jika ditemukan perbedaan antara teks dengan pattern (*mismatch*).

Pada algoritma ini konsep *prefix* dan *suffix* digunakan untuk menentukan berapa banyak pergeseran yang perlu dilakukan. Pergeseran dilakukan sebanyak mungkin sedemikian sehingga tidak perlu mengulang pemeriksaan yang sama. Jika terjadi *mismatch* teks T pada $T[i]$ dan pattern P pada $P[j]$ ($T[i] \neq P[j]$) maka jumlah pergeseran yang perlu dilakukan adalah sebanyak nilai *prefix* $P[0..j-1]$ terbesar yang juga merupakan *suffix* dari $P[1..j-1]$.

Algoritma KMP akan melakukan preproses terhadap pattern P yang akan dicari kecocokannya pada teks T dengan mencari semua nilai *prefix* terbesar pada $P[0..j-1]$ yang sama dengan *suffix* pada $P[1..j-1]$ jika terjadi *mismatch* pada $P[j]$. *Border function* $b(k)$ adalah ukuran terbesar pada $P[0..k]$ yang juga merupakan *suffix* dari $P[1..k]$. Dengan demikian langkah-langkah pencocokan string dengan algoritma KMP adalah sebagai berikut.

1. Input teks dan *pattern*
2. Hitung *border function* untuk setiap nilai j *pattern*
3. Lakukan pencocokan string yang apabila jika terjadi *mismatch* maka nilai j yang baru ditentukan dari nilai *border function* yang telah dihitung sebelumnya.

Kompleksitas total pencocokan string dengan algoritma KMP yaitu $O(m+n)$. Algoritma ini cocok untuk memproses file ukuran besar. Namun ketika digunakan pada teks yang beragam akan sering terjadi *mismatch* sehingga prosesnya bisa menjadi seperti *brute force*.

2. Algoritma Boyer Moore

Algoritma Boyer Moore adalah salah satu algoritma pencocokan string yang paling efektif saat ini. Algoritma Boyer Moore terdiri dari 2 teknik yaitu *the looking glass technique*

(pencocokan string dilakukan seolah-olah terbalik: dimulai dari indeks terakhir pattern P) dan *the character jump technique* (ketika terjadi *mismatch* $P[j] \neq T[i]$ dengan $T[i] = x$ akan dilakukan lompatan pada posisi tertentu). Dalam menentukan posisi lompatan pada Algoritma Boyer Moore terdapat 3 kasus yang perlu diperhatikan yaitu sebagai berikut.

1. Apabila terjadi *mismatch* pada $T[i]$ dan $P[j]$, panjang $P = m$, karakter $T[i]$ adalah x , dan terdapat x di P dengan indeks yang lebih kecil daripada j maka seolah-olah P digeser ke kanan agar posisi x di $T[i]$ sejajar dengan posisi kemunculan terakhir(lo) x di P ($i' = i + (m-1)-lo$)
2. Apabila terjadi *mismatch* pada $T[i]$ dan $P[j]$, panjang $P = m$, karakter $T[i]$ adalah x , dan terdapat x di P dengan indeks yang lebih besar daripada j maka seolah-olah P digeser 1 karakter ke kanan ($i' = i+m-j$)
3. Apabila terjadi *mismatch* pada $T[i]$ dan $P[j]$, panjang $P = m$, karakter $T[i]$ adalah x , tapi tidak terdapat x di P maka seolah-olah P digeser agar $P[0]$ sejajar dengan $T[i+1]$ ($i' = i+m$)

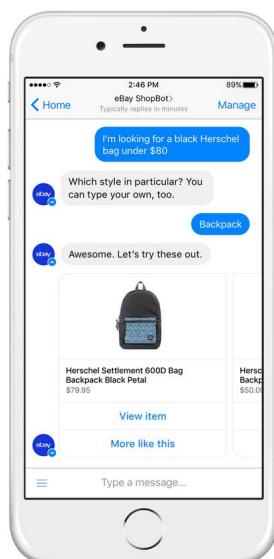
Pada Algoritma Boyer Moore diperlukan fungsi last occurrence $L(x)$ untuk menentukan posisi kemunculan terakhir suatu karakter pada P . Oleh karena itu sebelum melakukan pencocokan string terlebih dahulu tentukan posisi kemunculan terakhir semua karakter pada teks T di dalam pattern P . Jika suatu karakter x pada T tidak muncul di P maka $L(x) = -1$. Kompleksitas total pencocokan string dengan algoritma Boyer Moore untuk kasus terburuk yaitu $O(nm+A)$. Algoritma Boyer Moore ini cepat untuk teks dengan variasi karakter yang beragam, namun lambat untuk teks dengan variasi karakter tidak beragam.

3. Regular Expression (Regex)

Regular expression (regex) adalah cara mendeskripsikan suatu bahasa secara aljabar. Regex merupakan sebuah teks (string) yang mendefinisikan sebuah pola pencarian sehingga dapat membantu kita untuk melakukan *matching* (pencocokan) atau *locate* (pencarian) teks. Dengan kata lain kita dapat melakukan pencarian pada teks dengan menggunakan regular expression. Setiap karakter pada regular expression memiliki arti tertentu, misalnya ‘.’ untuk menyatakan semua karakter kecuali karakter baris baru ('\n'), ‘^’ untuk menyatakan awal string, ‘[abc]’ untuk menyatakan karakter ‘a’, ‘b’, atau ‘c’, ‘+’ untuk menyatakan karakter sebelumnya muncul minimal 1 kali, ‘|’ untuk atau, ‘\d’ untuk mendeskripsikan angka, ‘\w’ untuk mendeskripsikan karakter huruf dan angka, dan sebagainya.

4. Chatbot

Chatbot adalah sebuah perangkat lunak percakapan yang didukung oleh respons yang telah diprogram sebelumnya atau kecerdasan buatan (AI) untuk menjawab pertanyaan tanpa memerlukan operator manusia. Chatbot pertama diciptakan oleh Joseph Wiesenbaum pada tahun 1966 yang dinamai Eliza. Walaupun sudah ada sejak lama chatbot masih populer hingga sekarang karena manfaatnya. Ada 3 macam metode sistem operasional yang digunakan dalam chatbot, yaitu *pattern matching* (mencari kata kunci yang dimasukkan pengguna, kemudian akan ditanggapi dengan jawaban yang paling sesuai dengan kata kunci yang dipilih), *decision tree-based* (pengguna hanya dapat memilih pertanyaan yang telah disediakan oleh sistem), dan metode kontekstual (mengandalkan pembelajaran mesin untuk bisa menciptakan percakapan secara natural).



Gambar: contoh chatbot

Sumber: [10 Exemples de Chatbots à utiliser pour son Business | Alioze](#)

BAB III

Analisis Pemecahan Masalah

1. Langkah-Langkah Pemecahan Masalah

1. Menambahkan *task* baru

Untuk menambahkan *task* baru, pertama-tama dilakukan pengecekan apakah masukan pengguna merupakan *command* yang sesuai untuk menambahkan *task* baru, yaitu *command* yang di dalamnya terdapat tanggal dengan format DD/MM/YYYY, kode mata kuliah dengan format HHAAAA (2 huruf 4 angka), jenis tugas berdasarkan kata penting, dan topik tugas. Tanggal, kode mata kuliah, jenis tugas, dan topik tugas diidentifikasi menggunakan *regular expression* melalui *pattern matching*. Apabila *command* tersebut sesuai, maka dilakukan penambahan *task* ke basis data, kemudian ditampilkan pesan berhasil apabila penambahan ke basis data berhasil dilakukan.

2. Melihat daftar *task* yang harus dikerjakan

Untuk melihat daftar task yang harus dikerjakan, pertama-tama dilakukan pengecekan apakah *command* yang dimasukkan oleh pengguna adalah *command* yang memadai untuk melihat daftar task, yaitu yang memenuhi salah satu di antara pilihan berikut.

a. Melihat daftar *task* untuk semua waktu

Memuat kombinasi kata penting yaitu “deadline” atau “kuis” atau “uts” atau “uas” atau “tubes” atau “tucil” atau “pr” yang diidentifikasi menggunakan *regular expressions* dan juga kata “sejauh ini” atau “apa saja” yang diidentifikasi secara *exact matching* menggunakan algoritma KMP dan bersifat *case-insensitive*.

b. Melihat daftar *task* yang jatuh tempo di antara dua tanggal

Memuat dua tanggal dengan format DD/MM/YYYY yang diidentifikasi menggunakan *regular expression*. Selain itu, masukan dari pengguna juga harus memuat kata “deadline” atau “kuis” atau “uts” atau “uas” atau “tubes” atau “tucil” atau “pr”. Pengidentifikasian kata menggunakan *regular expression* dan bersifat *case-insensitive*.

c. Melihat daftar task untuk hari ini

Memuat kata “hari ini” yang diidentifikasi menggunakan algoritma KMP, serta memuat kata “deadline” atau “kuis” atau “uts” atau “uas” atau “tubes” atau “tucil” atau “pr” yang diidentifikasi menggunakan *regular expression*.

d. Melihat daftar *task* yang jatuh tempo beberapa hari ke depan

Memuat klausanya dengan format “[N] hari ke depan” dengan N merupakan angka. Identifikasi format ini dilakukan melalui *pattern matching* menggunakan *regular expression*. Selain itu, masukan harus memuat kata “deadline” atau “kuis” atau “uts” atau “uas” atau “tubes” atau “tucil” atau “pr” yang juga diidentifikasi menggunakan *regular expression*.

e. Melihat daftar *task* yang jatuh tempo beberapa minggu ke depan

Memuat klausanya dengan format “[N] minggu ke depan” dengan N merupakan angka. Identifikasi format ini dilakukan melalui *pattern matching* menggunakan *regular expression*. Selain itu, masukan harus memuat kata “deadline” atau “kuis” atau “uts” atau “uas” atau “tubes” atau “tucil” atau “pr” yang juga diidentifikasi menggunakan *regular expression*.

Apabila masukan sesuai, maka bot akan menampilkan daftar *task* dari basis data yang sesuai dengan ketentuan yang dimuat di dalam *command*.

3. Menampilkan *deadline* dari suatu *task* tertentu

Untuk menampilkan *deadline* dari suatu *task* tertentu, pertama-tama dilakukan pengecekan apakah masukan pengguna merupakan *command* yang sesuai, yaitu memuat kata “kapan” dan “deadline” yang diidentifikasi menggunakan *exact matching* algoritma KMP yang bersifat *case-insensitive*, serta memuat kode mata kuliah yang diidentifikasi menggunakan *regular expression*. Apabila *command* sesuai, dilakukan pencarian kode mata kuliah tersebut di dalam basis data yang kemudian tanggal *deadline*-nya ditampilkan ke layar pengguna.

4. Memperbarui task tertentu

Untuk memperbarui *task* tertentu pertama-tama dilakukan pengecekan apakah masukan pengguna sesuai untuk *command* memperbarui tanggal suatu *task*, yaitu memiliki setidaknya salah satu kata kunci ‘*jadi*’, ‘*undur*’, ‘*maju*’, ID *task* dengan format task X dan juga tanggal dengan format DD/MM/YYYY. Pengecekan kata kunci ‘*jadi*’, ‘*undur*’, ‘*maju*’ dilakukan dengan menggunakan algoritma KMP

sedangkan pengecekan tanggal dan ID *task* dilakukan dengan menggunakan *regular expression*. Apabila *command* sesuai, maka akan dilakukan update tanggal task ke basis data, kemudian akan ditampilkan pesan berhasil memperbarui task dari bot jika proses update task ke basis data berhasil dan akan menampilkan pesan gagal jika task tidak dikenali.

5. Menandai bahwa suatu task sudah selesai dikerjakan

Untuk menandai bahwa suatu *task* sudah selesai, pertama-tama dilakukan pengecekan apakah masukan pengguna sesuai untuk *command* memperbarui tanggal suatu *task*. Untuk melakukan pengecekan ini semua huruf di input diubah menjadi huruf kecil, kemudian dilakukan pengecekan apakah masukan memiliki setidaknya salah satu dari kata kunci ‘*jadi*’, ‘*undur*’, ‘*maju*’, terdapat task beserta nomornya di input, dan terdapat tanggal dengan format lengkap di input. Pengecekan kata kunci ‘*jadi*’, ‘*undur*’, ‘*maju*’ dilakukan dengan menggunakan algoritma KMP sedangkan tanggal dan task beserta nomornya dicek dengan menggunakan regular expression. Apabila *command* tersebut memadai maka akan dilakukan update tanggal task ke basis data, kemudian akan ditampilkan pesan berhasil memperbarui task dari bot jika proses update task ke basis data berhasil dan akan menampilkan pesan gagal jika task tidak dikenali.

6. Menampilkan opsi *help* yang difasilitasi oleh asisten

Untuk dapat menampilkan opsi *help*, pertama-tama dilakukan pengecekan apakah masukan pengguna sesuai, yaitu memuat kata “*apa*”, “*bisa*”, dan “*bot*”. Pencocokan dilakukan menggunakan algoritma KMP dan bersifat *case-insensitive*. Apabila *command* sesuai, ditampilkan teks berisi fitur-fitur yang bisa diakses melalui bot, daftar kata penting, dan daftar format yang perlu diperhatikan.

7. Menampilkan pesan *error* jika bot tidak dapat mengenali masukan pengguna

Apabila masukan pengguna tidak memenuhi kriteria *command* untuk melaksanakan keenam poin di atas, maka bot akan menampilkan pesan *error* ke layar pengguna yang menunjukkan bahwa masukan pengguna tidak dikenali.

2. Fitur Fungsional dan Arsitektur Chatbot

1. Fitur Fungsional Chatbot

Fitur-fitur yang dimiliki oleh AntiKeosKeosBot adalah sebagai berikut.

a. Menambahkan *task* baru

Menambahkan *task* baru ke dalam basis data yang memuat informasi jenis *task*, kode mata kuliah (HHAAAA: 2 huruf 4 angka), topik, dan juga tanggal jatuh tempo (DD/MM/YYYY).

b. Melihat daftar *task* yang harus dikerjakan

- i. Melihat daftar *task* untuk semua waktu, baik seluruh jenis *task* maupun jenis *task* tertentu (berdasarkan kata penting pada masukan pengguna, yaitu “deadline”, “kuis”, “uts”, “uas”, “tubes”, “tucil”, “pr”)
- ii. Melihat daftar *task* yang jatuh tempo di antara dua tanggal, dengan kedua tanggal diperoleh dari masukan pengguna dengan format DD-MM-YYYY, baik untuk seluruh jenis *task* maupun jenis *task* tertentu (berdasarkan kata penting pada masukan pengguna, yaitu “deadline”, “kuis”, “uts”, “uas”, “tubes”, “tucil”, “pr”)
- iii. Melihat daftar *task* yang jatuh tempo hari ini, baik seluruh jenis *task* maupun jenis *task* tertentu (berdasarkan kata penting pada masukan pengguna, yaitu “deadline”, “kuis”, “uts”, “uas”, “tubes”, “tucil”, “pr”)
- iv. Melihat daftar *task* yang jatuh tempo beberapa minggu ke depan, dengan jumlah hari tergantung masukan pengguna, baik untuk seluruh jenis *task* maupun jenis *task* tertentu (berdasarkan kata penting pada masukan pengguna, yaitu “deadline”, “kuis”, “uts”, “uas”, “tubes”, “tucil”, “pr”)
- v. Melihat daftar *task* yang jatuh tempo beberapa hari ke depan, dengan jumlah minggu tergantung masukan pengguna, baik untuk seluruh jenis *task* maupun jenis *task* tertentu (berdasarkan kata penting pada masukan pengguna, yaitu “deadline”, “kuis”, “uts”, “uas”, “tubes”, “tucil”, “pr”)

c. Menampilkan *deadline* suatu *task* tertentu

Menampilkan tanggal jatuh tempo suatu *task* dengan kode mata kuliah berdasarkan masukan pengguna (dengan format HHAAAA: 2 huruf 4 angka).

- d. Menandai suatu *task* sudah dikerjakan

Menghapus *task* dari basis data dengan ID *task* diperoleh berdasarkan masukan pengguna (dengan format task X, X merupakan ID *task*)

- e. Menampilkan opsi *help*

Menampilkan fitur-fitur yang bisa diakses pada AntiKeosKeosBot, menampilkan daftar kata penting, dan juga format penulisan tertentu untuk memudahkan pengguna.

- f. Menampilkan pesan error

Pesan error akan ditampilkan apabila input dari pengguna tidak dapat diterjemahkan menjadi command apapun.

2. Arsitektur Chatbot

Chatbot merupakan program berbasis web lokal yang menggunakan bahasa pemrograman Python dan *framework* Flask. Program ini terbagi ke dalam tiga file, yaitu sebagai berikut.

- a. utility.py, berisi method-method yang mendukung pencocokan string, yaitu untuk pemrosesan *regular expression* dan algoritma KMP, mengambil beberapa kata penting dari masukan pengguna, serta mengecek apakah masukan pengguna sesuai dengan pola untuk menjalankan *command* tertentu
- b. inoutput.py, berisi method-method yang melakukan pemrosesan input dan output yang juga menghubungkan antara masukan dari pengguna dan juga basis data.
- c. app.py, merupakan modul utama (*main*) yang menyambungkan *backend* dengan *frontend* sehingga menghasilkan sebuah web AntiKeosKeosBot yang utuh. Secara singkat, fitur pada web lokal yang dihasilkan adalah sebagai berikut.
 - Menu AntiKeosKeosBot, merupakan halaman utama sebagai tempat pengguna berkomunikasi dengan bot. Pada halaman ini terdapat *chatbox* sebagai tempat mengirim masukan, dan juga ditampilkan *history* percakapan antara pengguna dengan bot.
 - Menu How to use, merupakan halaman yang menampilkan cara menggunakan AntiKeosKeosBot

- Menu Credits, menampilkan *drop down* yang menampilkan nama-nama dari pembuat program.

Porgram AntiKeosKeosBot memanfaatkan basis data dengan menggunakan DBMS MySQL. Basis data bernama AntiKeosKeosBot dan disimpan pada file AntiKeosKeosBot.sql dengan struktur sebagai berikut.

- a. Tabel Task, dengan atribut sebagai berikut.
 - i. id : integer (*primary key*)
 - ii. tanggal : date
 - iii. kodeMatkul : varchar(6)
 - iv. jenis : varchar(255)
 - v. judul : varchar(255)
- b. Tabel Chat, dengan atribut sebagai berikut.
 - i. id : integer (*primary key*)
 - ii. text : varchar(1000)
 - iii. source : varchar(10)
 - iv. timeStamp : time

BAB IV

Implementasi dan Pengujian

1. Implementasi Program

- *Pseudocode program utama*

Program utama AntiKeosKeosBot ini secara garis besar adalah sebagai berikut.

```
ga ngerti :(
index() → render_template
{ me-render file index.html untuk ditampilkan pada browser, pada
fungsi ini juga terdapat prosedur untuk menangani method 'POST',
yakni saat pengguna men-submit query berupa kalimat }

how_to_use() → render_template
{ me-render file howtouse.html untuk ditampilkan pada browser, pada
fungsi ini juga terdapat prosedur untuk menangani method 'POST',
yakni saat pengguna men-submit query untuk kembali ke index.html }

preprosesChats(chats: list of tuple) → list of dictionary
{ berfungsi mengolah input hasil fetch oleh cursor menjadi list of
dictionary, di mana setiap dictionary mewakili satu entitas pada
tabel chats di database. Selain itu nilai key text juga diproses
menjadi kalimat per baris }

splitText(text: string) → list of string
{ melakukan split pada string sehingga menjadi list of string agar
dapat dicetak pada html }
```

- *fungsi dan prosedur dalam menambahkan task baru*

```
isInputCommand(in input:string) → boolean
{mengecek apakah input adalah command untuk menambahkan task}

inputCommand(in input: string) → string
{mengembalikan string yang ditampilkan bot jika input merupakan
command untuk menambahkan task
asumsi sudah diperiksa dengan isInputCommand(input)}
```

- *fungsi dan prosedur dalam menampilkan daftar deadline*

```
isAllCommand(in input: string) → boolean
{mengembalikan true apabila command menanyakan semua deadline}

allDeadline() → string
{mengembalikan string yang berisi daftar seluruh task dari database}

allDeadlineType(in type: string) → string
```

```

{mengembalikan string yang berisi daftar seluruh task dengan jenis tertentu dari database}

isPeriodCommand(in input: string) → boolean
{mengembalikan true apabila command menanyakan deadline di antara dua tanggal}

periodType(in type,tgl1,tgl2 : string) → string
{mengembalikan string yang berisi daftar task tipe tertentu dengan deadline antara tanggal1 hingga tanggal2}

periodDeadline(in tgl1,tgl2 : string) → string
{mengembalikan string yang berisi daftar task dengan deadline antara tanggal1 hingga tanggal2}

isWeeksCommand(in input : string) → boolean
{mengembalikan true apabila command menanyakan deadline N hari ke depan}

weeksDeadline(in N: int) → string
{mengembalikan string yang berisi daftar task dengan deadline hari ini hingga N minggu ke depan}

weeksDeadlineType(in N: int,in type: string) → string
{mengembalikan string yang berisi daftar task dengan deadline hari ini hingga N minggu ke depan dengan jenis task tertentu}

isDaysCommand(in input : string) → boolean
{mengembalikan true apabila command menanyakan deadline N hari ke depan}

daysDeadline(in N: int) → string
{mengembalikan string yang berisi daftar task dengan deadline hari ini hingga N hari ke depan}

daysDeadlineType(in N: int,in type: string) → string
{mengembalikan string yang berisi daftar task dengan deadline hari ini hingga N hari ke depan dengan jenis task tertentu }

isTodayCommand(in input : string) → boolean
{mengembalikan true apabila command menanyakan deadline hari ini}

todayDeadline(in N: int)→ string
{mengembalikan string yang berisi daftar task yang deadline-nya hari ini}

todayDeadlineType(in N: int,in type: string)→ string
{mengembalikan string yang berisi daftar task yang deadline-nya hari ini dengan jenis task tertentu}

```

- *fungsi dan prosedur untuk menampilkan deadline dari suatu task tertentu*

```

isKapanCommand(in input: string)→ boolean
{mengecek apakah input adalah command untuk menampilkan deadline task tertentu}

```

```
deadlineOneTask(in input: string) → string
{mengembalikan string yang berisi deadline dari suatu task tertentu}
```

- *fungsi dan prosedur untuk memperbarui task tertentu*

```
isUpdateCommand(in input: string) → boolean
{mengecek apakah input adalah command untuk mengupdate deadline
task}

updateCommand(in input: string) → string
{memperbarui tanggal dari task yang hendak diubah dan
mengembalikan pesan sukses/tidaknya perubahan}
```

- *fungsi dan prosedur untuk menandai bahwa suatu task sudah selesai dikerjakan*

```
isSelesaiCommand(in input: string) → boolean
{ mengecek apakah input adalah command untuk menghapus task yang
sudah selesai }

selesaiCommand(in input: string) → string
{mengembalikan string yang ditampilkan bot jika input merupakan
command untuk menghapus task yang sudah selesai}
```

- *fungsi dan prosedur untuk menampilkan opsi help yang difasilitasi oleh bot*

```
isHelpCommand(in input: string) → boolean
{mengembalikan true apabila command menanyakan hal yang bisa
dilakukan bot}

helpCommand(in input: string) → string
{mengembalikan string berisi panduan pengguna untuk mengakses fitur
bot}
```

- *fungsi dan prosedur untuk menampilkan pesan error*

```
nothingCommand() → string
{mengembalikan string berisi pesan error apabila command tidak
dikenali}
```

- *fungsi dan prosedur dalam string matching*

```
{== ALGORITMA STRING MATCHING KMP ==}
compPrefSuf(in word: string, in length: int) → boolean
{mengembalikan true apabila prefix dan sufix sepanjang length sama}

getPref(in word: string, in lenI int ) → string
{mengembalikan prefix sepanjang len dari sebuah string word}

getLPS(in pat: string) → int[]
```

```

{mengembalikan array yang memuat jumlah irisan terbesar antara
prefix dan suffix}

KMPMatch(in pattern, text : string ) → int
{mengembalikan indeks dari karakter pertama string matching antara
pattern dengan text}

{== PATTERN MATCHING DENGAN REGEX ==}
getMatkul(in command: string) → int/string
{mengembalikan kode mata kuliah yang terdapat pada command}

getJenis(in command: string) → int/string
{mengembalikan jenis task yang terdapat pada command}

getJudul(in command: string) → int/string
{mengembalikan topik dari task yang terdapat pada command}

getDate(in input: string) → string[]
{mengembalikan tanggal yang terdapat pada command dalam format
YYYY/MM/DD}

getDays(in command: string) → int
{mengembalikan N dari klausa "N hari ke depan"}

getWeeks(in command: string) → int
{ mengembalikan N dari klausa "N minggu ke depan"}

getID(in input: string) → int
{ mengembalikan ID task yang terdapat pada command}

```

- fungsi *outputBot(in input: string) → string*

```

outputBot(in input: string) → string
{fungsi ini akan dipanggil di app.py dengan parameter adalah
inputan user.
Fungsi ini kemudian melakukan pengecekan command mana yang akan
dijalankan berdasarkan masukan user
fungsi ini kemudian akan mengolah command yang sesuai
hasil yang akan dicetak oleh bot ke layar akan dikeluarkan oleh
fungsi ini dalam bentuk string}

```

- fungsi dan prosedur lainnya

```

dateDDMMYYYY(in date: string) → string
{mengembalikan date berformat DD/MM/YYYY dari masukan yang
YYYY/MM/DD}

```

2. Struktur Data

Struktur data yang digunakan pada AntiKeosKeosBot adalah sebagai berikut.

a. Chat

Merupakan struktur data yang menyimpan *chat* antara pengguna dengan bot, yang memiliki atribut sebagai berikut.

- i. id : penanda unik *chat*
- ii. text : pesan teks pada *chat*
- iii. source : pengirim dari *chat*, apakah pengguna atau bot
- iv. time : waktu pengiriman *chat*

3. Tata Cara Penggunaan Program

a. Kebutuhan Instalasi

Beberapa kebutuhan instalasi dalam menjalankan program AntiKeosKeosBot adalah sebagai berikut.

- i. Python versi 3.8 ke atas
- ii. MariaDB versi 10.5 ke atas
- iii. Library flask, dapat diinstal menggunakan *command*
`pip install flask`
- iv. Library mysql-connector, dapat diinstal menggunakan *command*
`pip install mysql-connector`

b. Persiapan Program

Persiapan yang dibutuhkan adalah dengan men-store AntiKeosKeosBot.sql ke dalam *database* lokal MariaDB. Langkah-langkahnya adalah sebagai berikut.

- i. Buka MariaDB, buat *database* baru menggunakan *command*
`create database MariaDB;`
- ii. Keluar dari MariaDB, arahkan direktori ke folder test, kemudian *store database* menggunakan *command*
`mysql -u {username} -p AntiKeosKeosBot < AntiKeosKeosBot.sql`

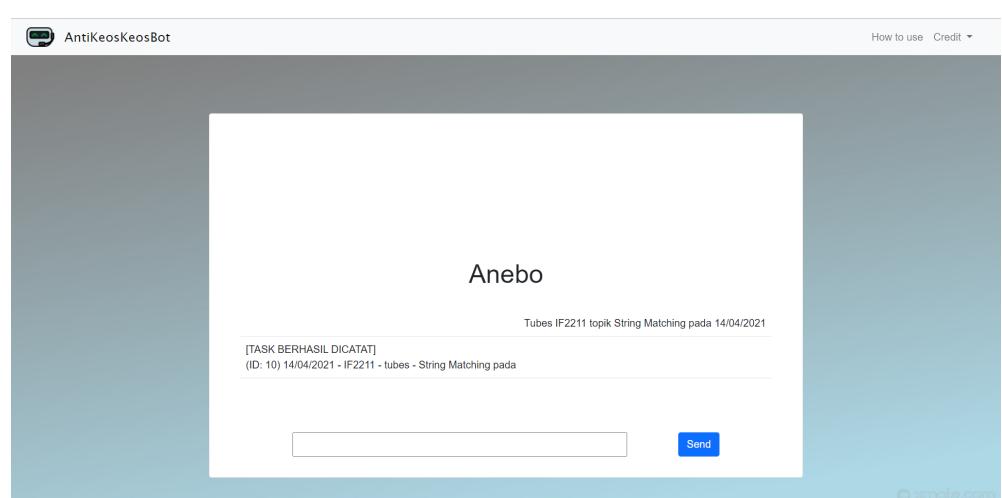
c. Cara Menjalankan

- i. Pertama-tama, buka file inoutput.py pada IDE, kemudian isi *password* pada *line 7* dengan *password* MariaDB

- ii. Pada folder tempat penyimpanan *repository*, buka *command prompt*, kemudian jalankan program dengan menggunakan *command python src/app.py*
 - iii. Tunggu hingga program men-generate link alamat lokal
 - iv. Buka link tersebut di *browser*
 - v. Program AntiKeosKeosBot siap untuk dikalankan
- d. Cara Menggunakan
- i. Klik tab AntiKeosKeosBot untuk masuk ke halaman utama dan berkomunikasi dengan bot
 - Masukkan input ke dalam *chatbox* yang tersedia kemudian klik tombol send
 - ii. Klik tab How to use untuk masuk ke halaman yang menampilkan cara penggunaan program
 - iii. Klik tab Credit untuk melihat *contact person*

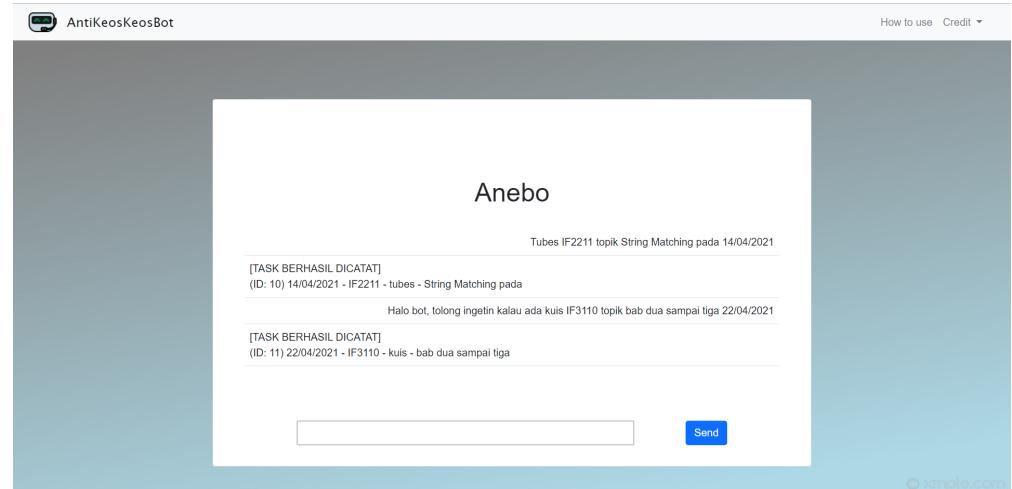
4. Hasil Pengujian

- a. Menambahkan task baru
- i. Masukan pengguna: Tubes IF2211 topik String Matching pada 14/04/2021
- Keluaran:

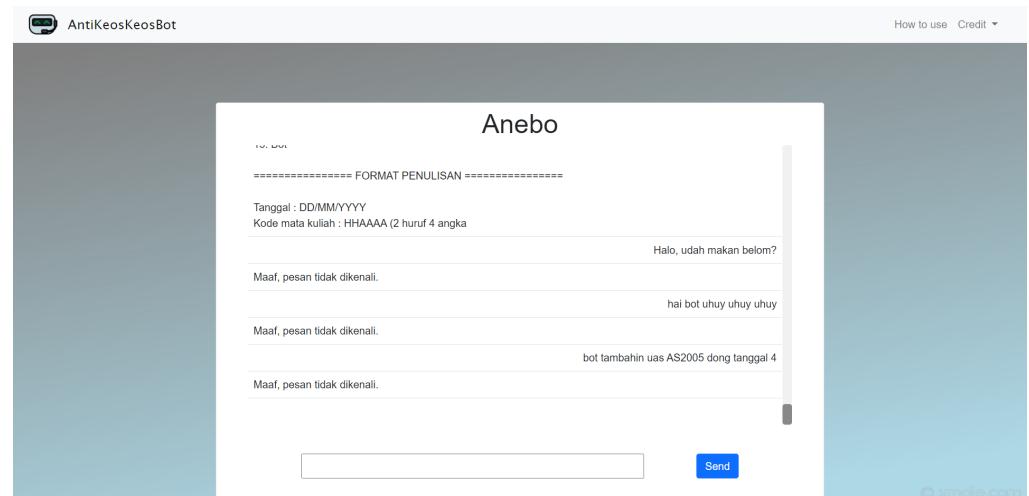


- ii. Masukan pengguna: Halo bot, tolong ingetin kalau ada kuis IF3110 topik bab dua sampai tiga 22/04/2021

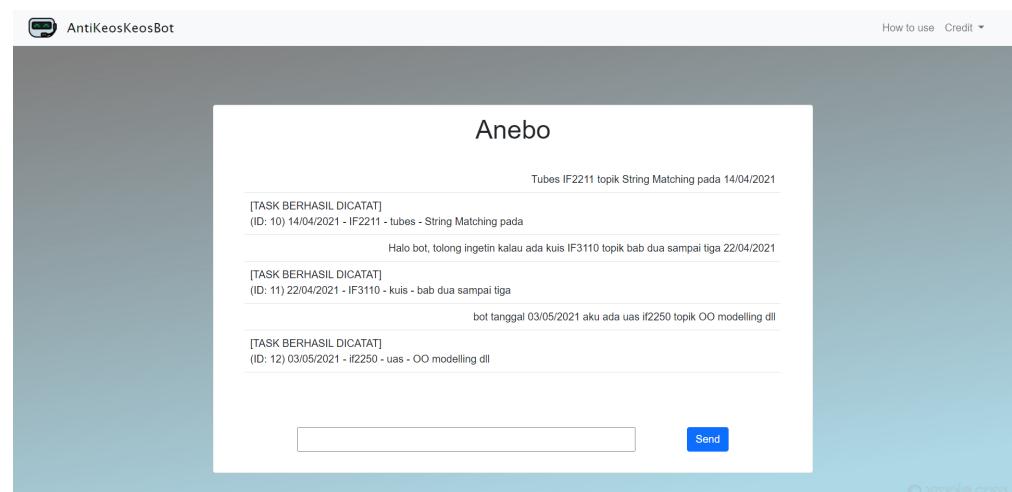
Keluaran:



- iii. Masukan pengguna: bot tambahin uas AS2005 dong tanggal 4
Keluaran:



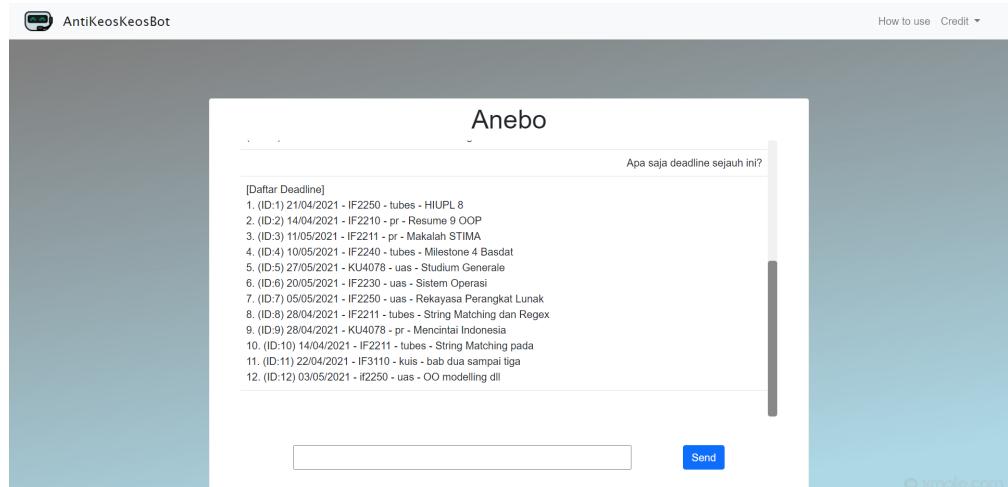
- iv. Masukan pengguna: bot tanggal 03/05/2021 aku ada uas if2250 topik OO
modelling dll
Keluaran:



b. Melihat daftar task yang harus dikerjakan

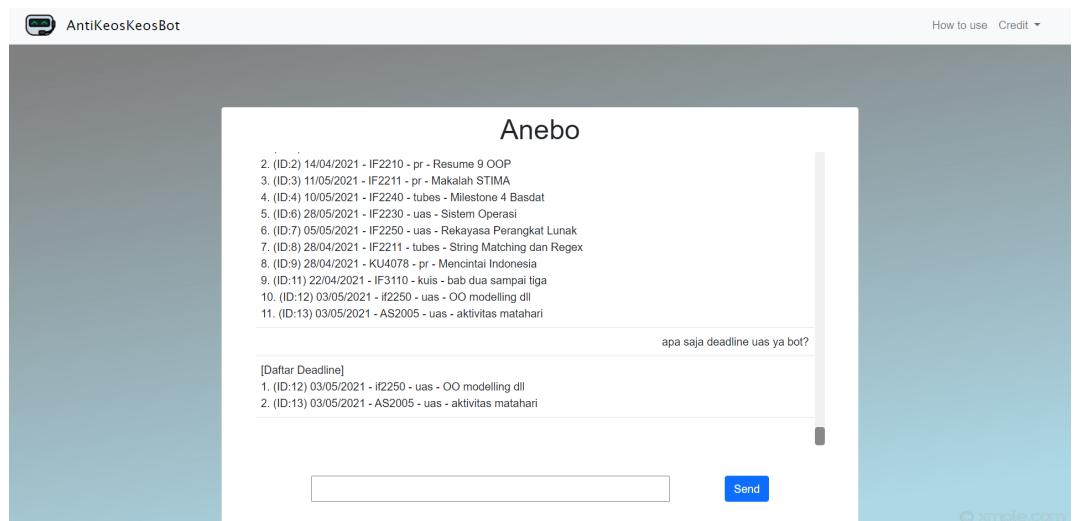
i. Masukan pengguna: Apa saja deadline sejauh ini?

Keluaran:



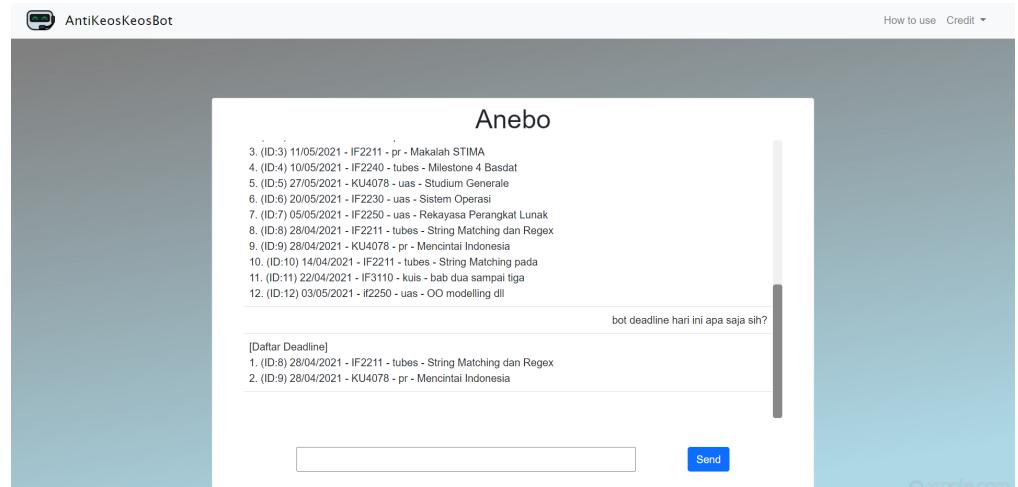
ii. Masukan pengguna: apa saja deadline uas ya bot?

Keluaran:



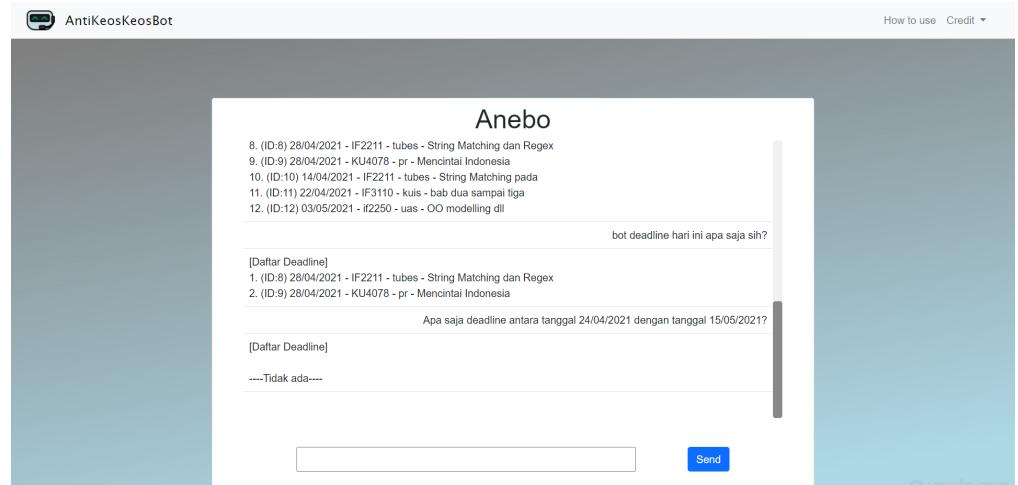
iii. Masukan pengguna: bot deadline hari ini apa saja sih?

Keluaran:

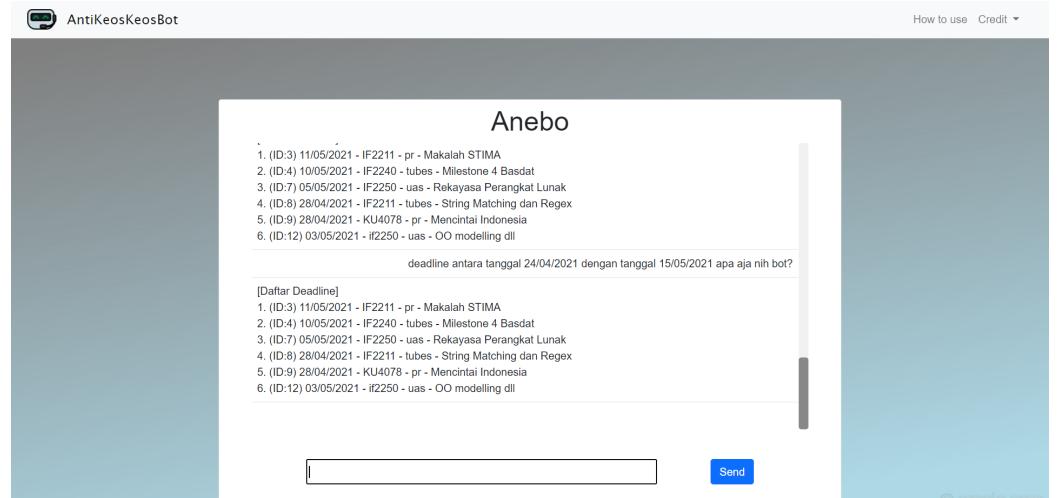


- iv. Masukan pengguna:Apa saja deadline antara tanggal 24/04/2021 dengan tanggal 15/05/2021?

Keluaran:

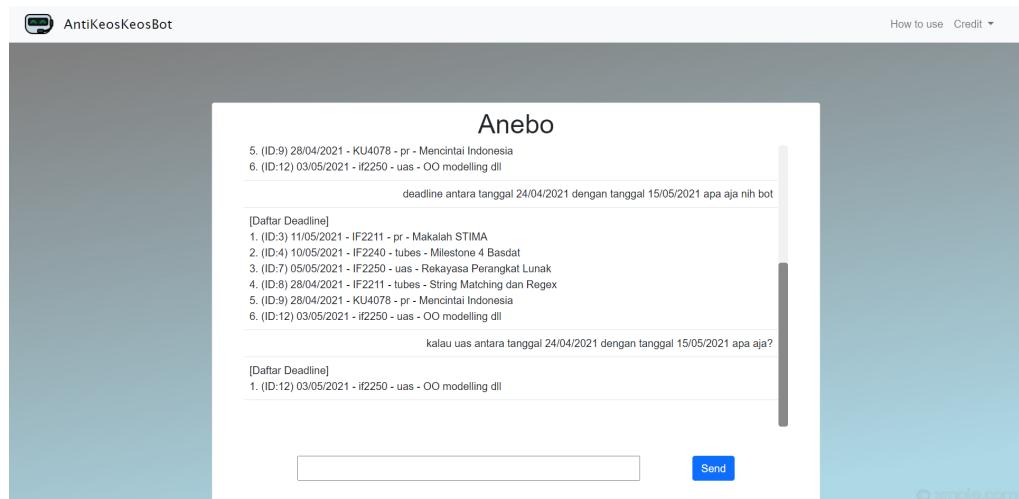


- v. Masukan pengguna: deadline antara tanggal 24/04/2021 dengan tanggal 15/05/2021 apa aja nih bot?



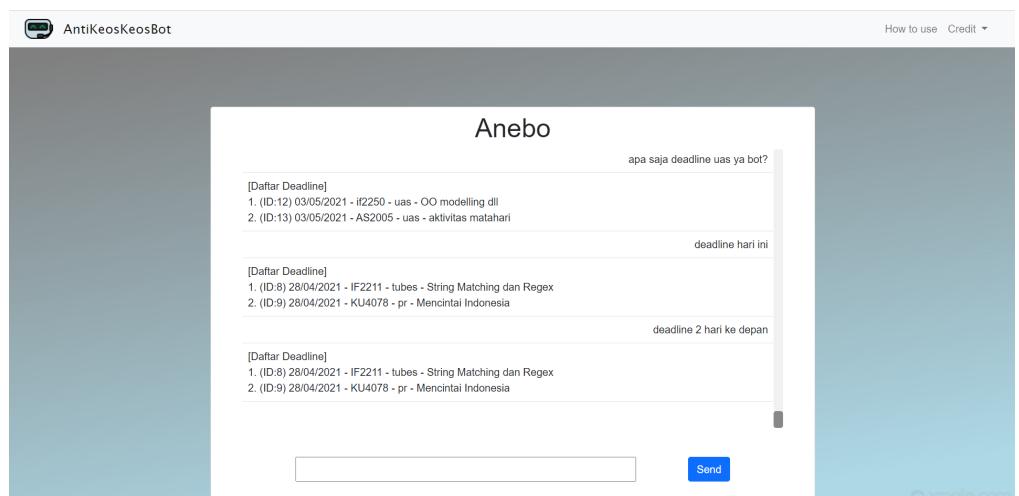
- vi. Masukan pengguna: kalau uas antara tanggal 24/04/2021 dengan tanggal 15/05/2021 apa aja?

Keluaran:



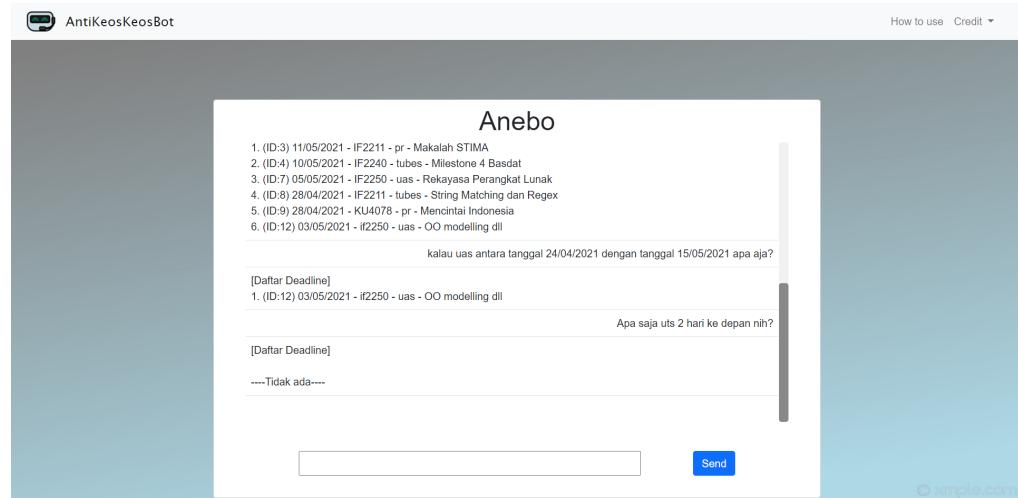
- vii. Masukan pengguna: deadline 2 hari ke depan

Keluaran:



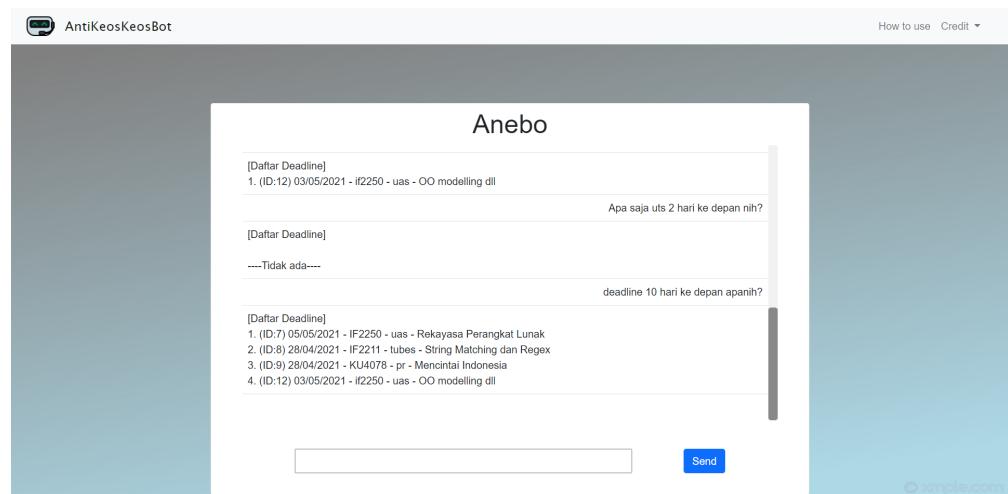
- viii. Masukan pengguna: Apa saja uts 2 hari ke depan nih?

Keluaran:



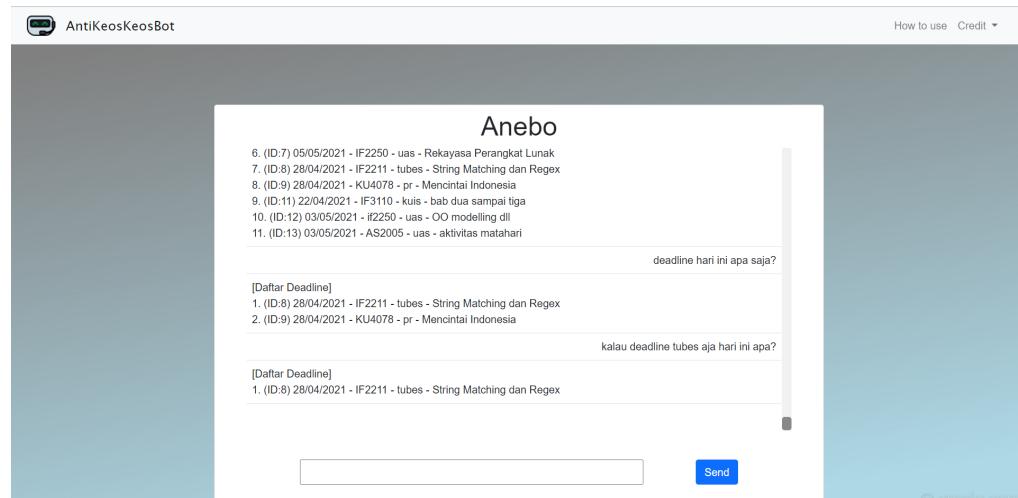
ix. Masukan pengguna: deadline 10 hari ke depan apanih?

Keluaran:



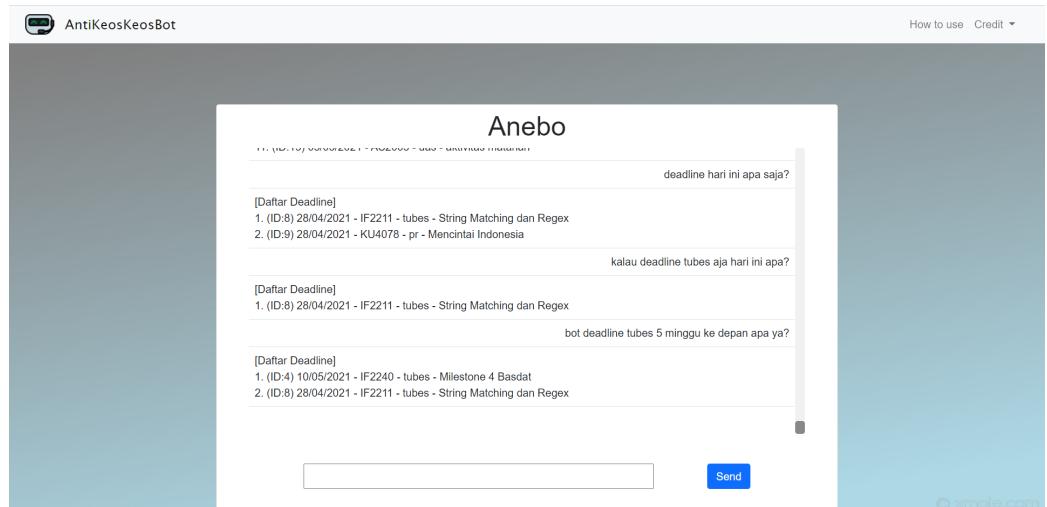
x. Masukan pengguna: kalau deadline tubes aja hari ini apa?

Keluaran:



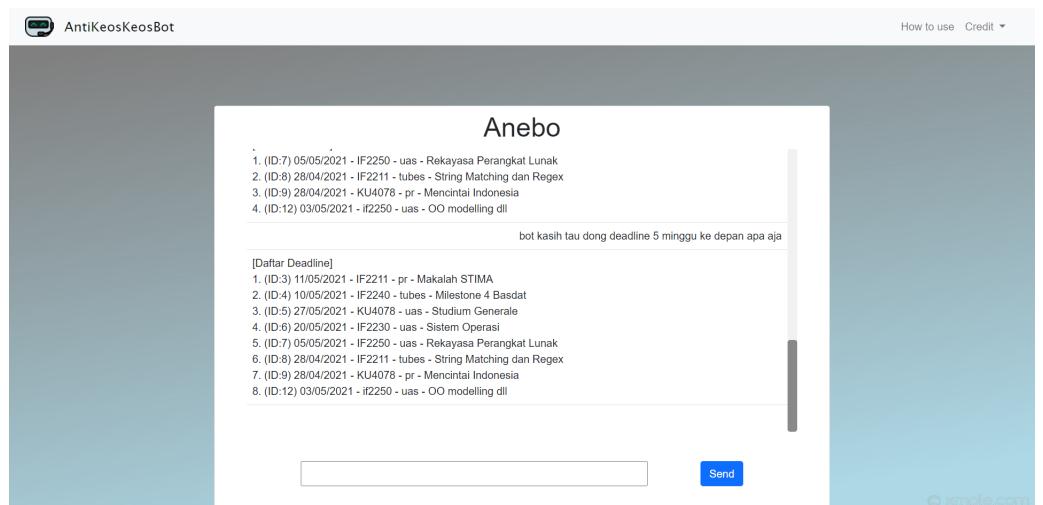
xi. Masukan pengguna: bot deadline tubes 5 minggu ke depan apa ya?

Keluaran:



xii. Masukan pengguna: bot kasih tau dong deadline 5 minggu ke depan apa aja

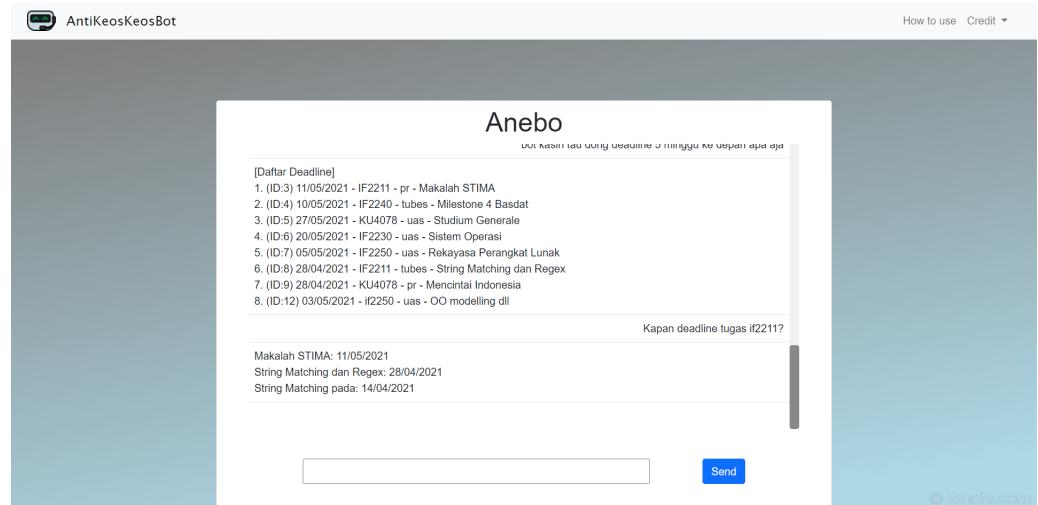
Keluaran:



c. Menampilkan deadline dari suatu task tertentu

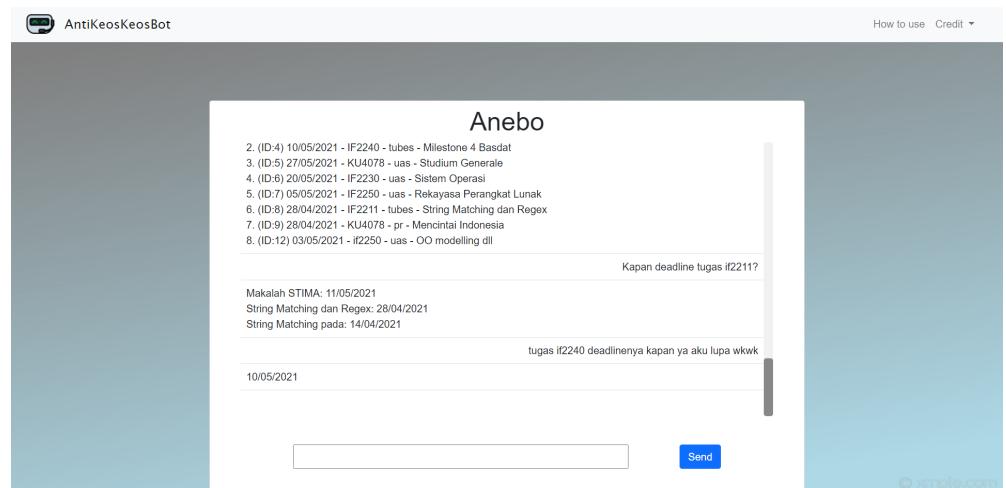
i. Masukan pengguna: Kapan deadline tugas if2211?

Keluaran:



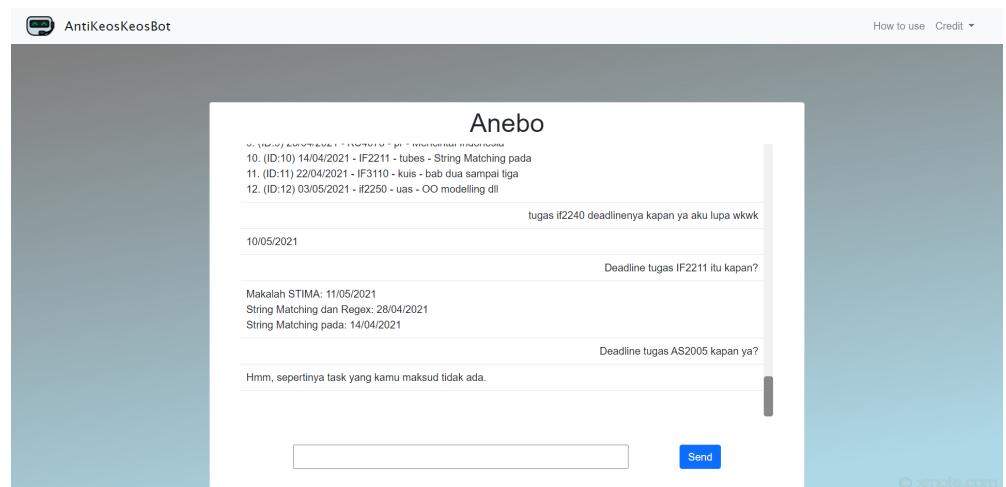
ii. Masukan pengguna: tugas if2240 deadlinenya kapan ya aku lupa wkwk

Keluaran:



iii. Masukan pengguna: Deadline tugas AS2005 itu kapan ya?

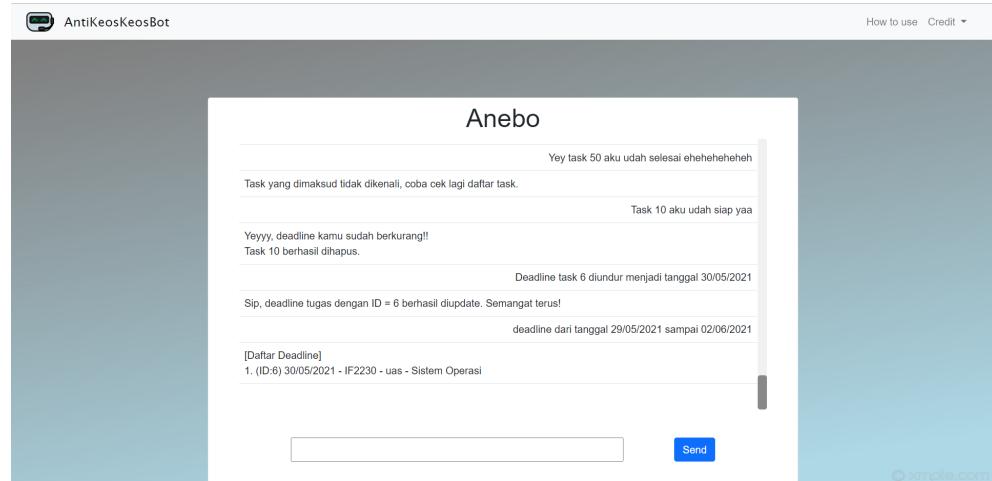
Keluaran:



d. Memperbarui task tertentu

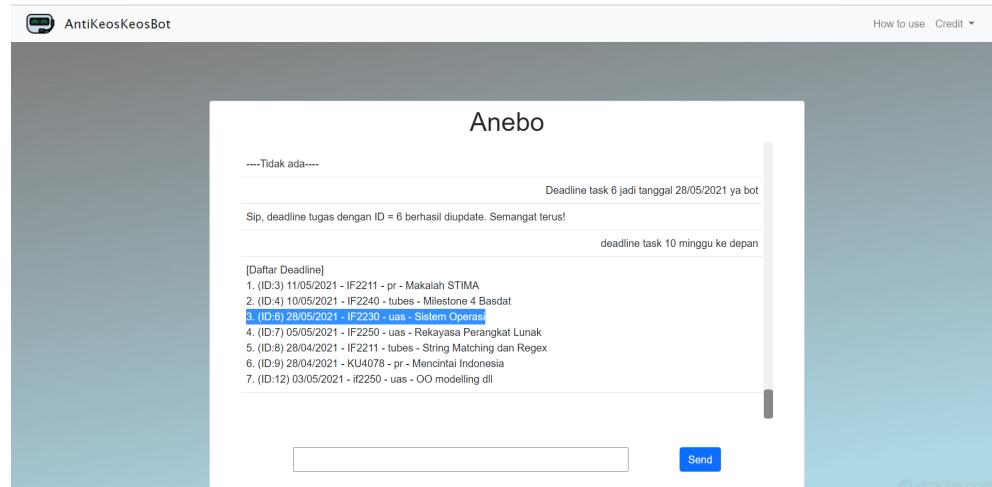
- i. Masukan pengguna:Deadline task 6 diundur menjadi tanggal 30/05/2021

Keluaran:



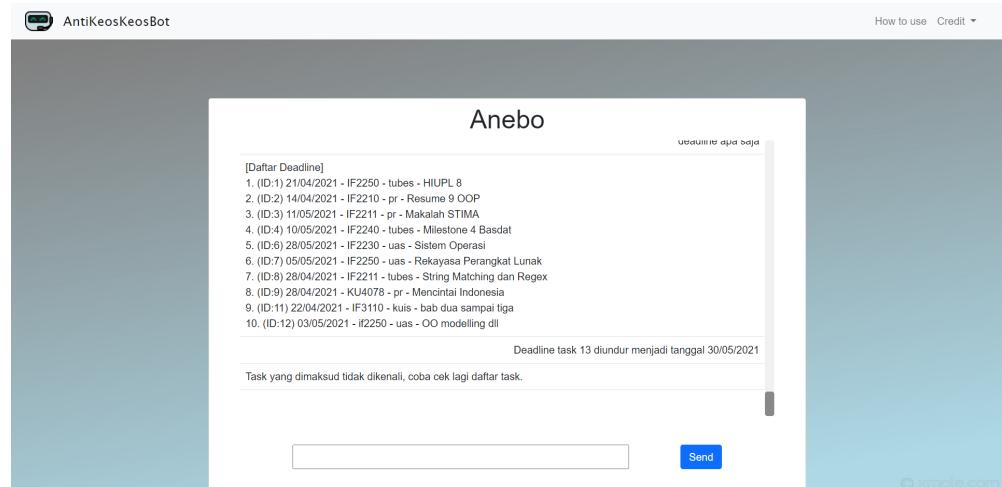
- ii. Masukan pengguna:Deadline task 6 jadi tanggal 28/05/2021 ya bot

Keluaran:



- iii. Masukan pengguna:Deadline task 13 diundur menjadi tanggal 30/05/2021

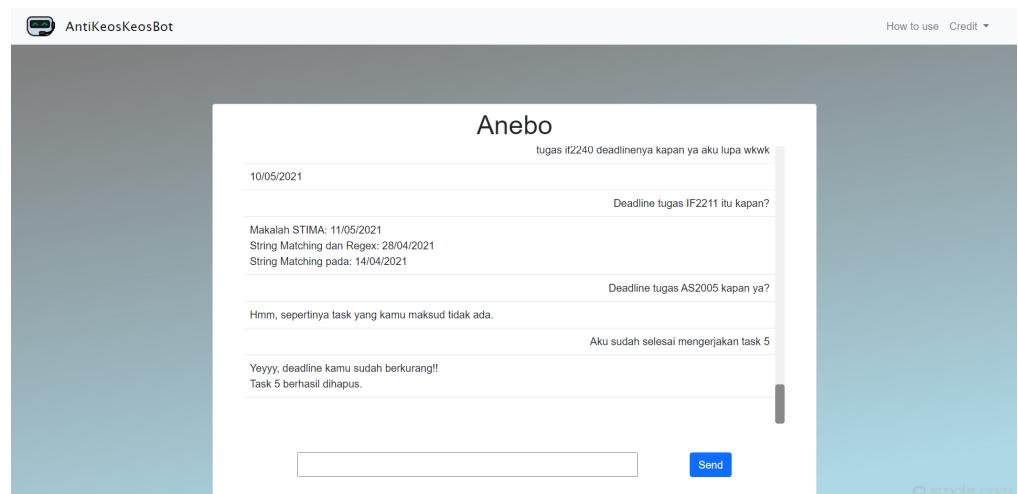
Keluaran:



e. Menandai bahwa suatu task sudah selesai dikerjakan

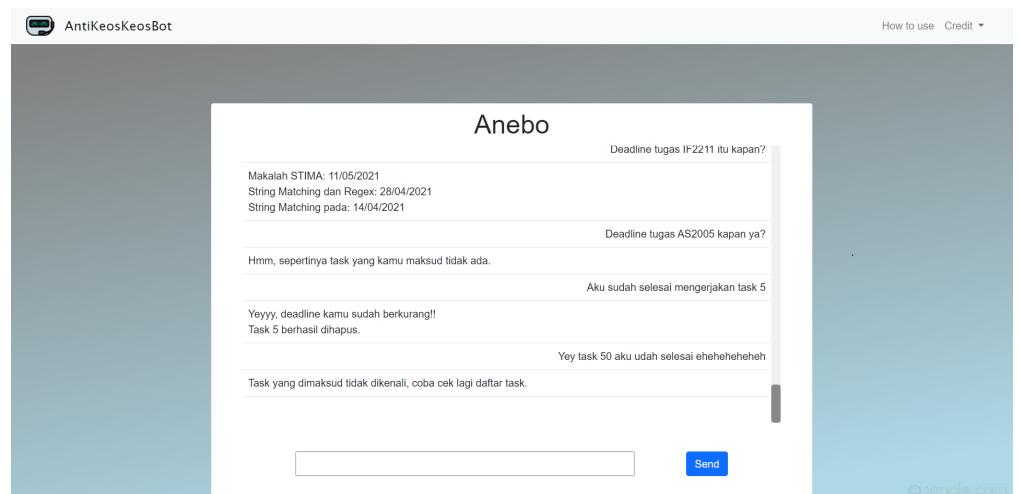
i. Masukan pengguna: Aku sudah selesai mengerjakan task 5

Keluaran:



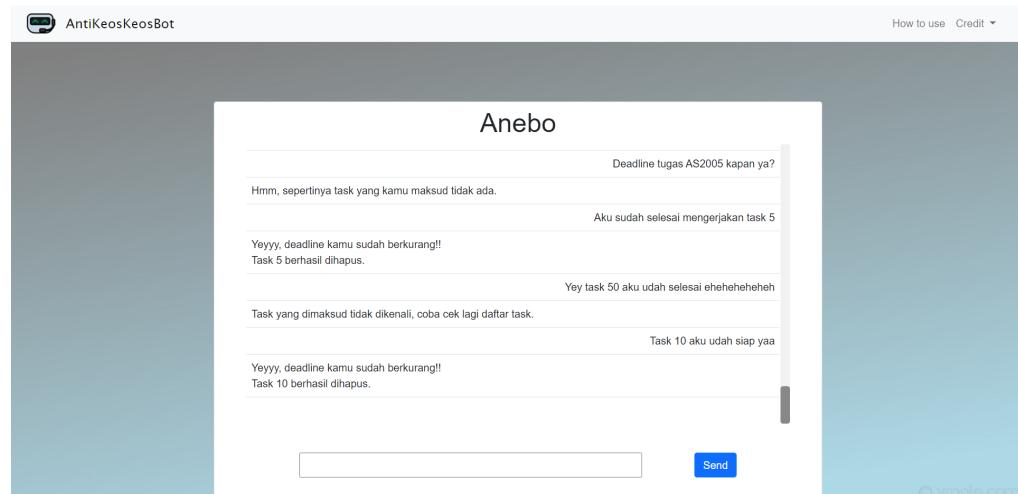
ii. Masukan pengguna: Yey task 50 aku udah selesai eheheheheh

Keluaran



iii. Masukan pengguna: Task 10 aku udah siap yaa

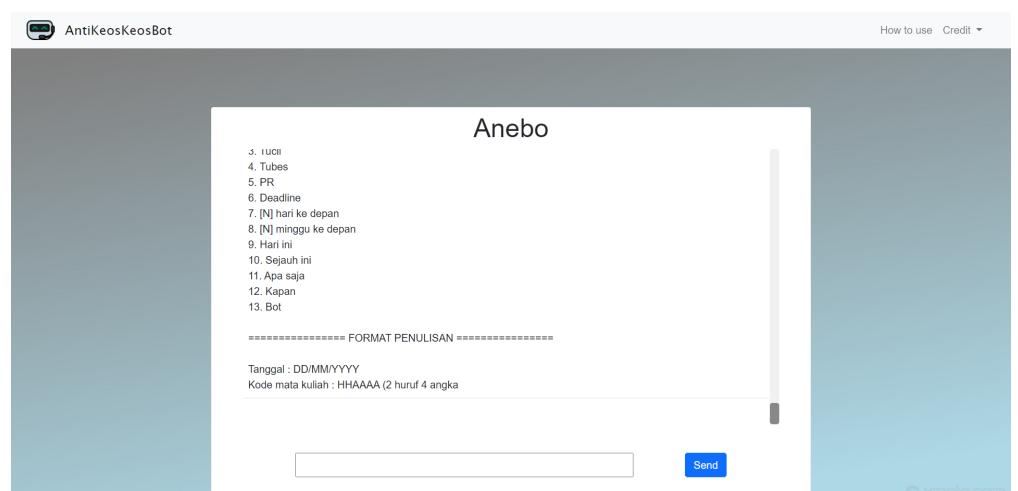
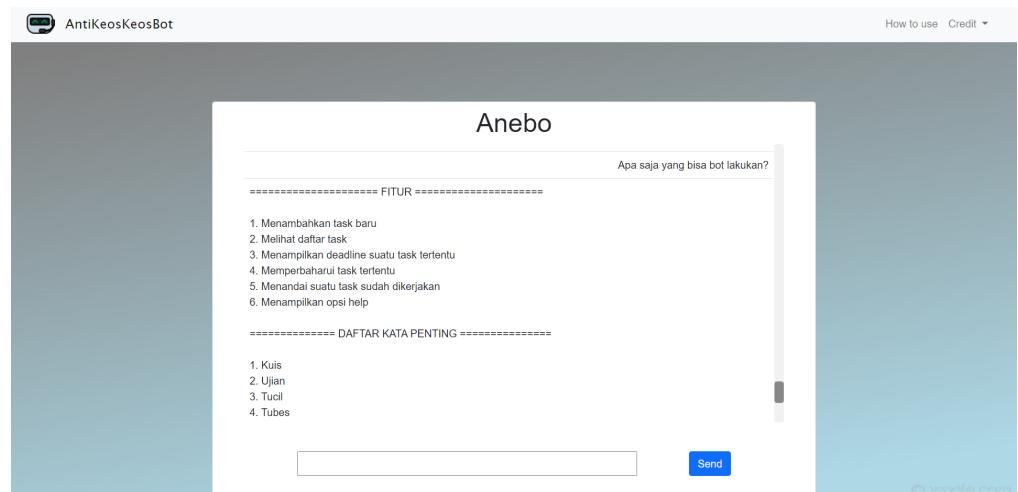
Keluaran



f. Menampilkan opsi *help* yang difasilitasi oleh bot

i. Masukan pengguna: Apa saja yang bisa bot lakukan?

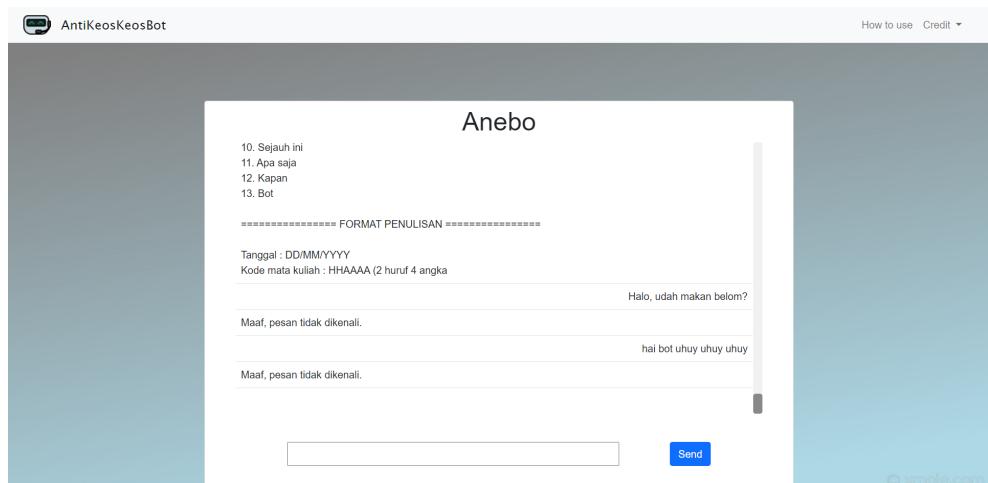
Keluaran:



g. Menampilkan pesan error apabila pesan tidak dikenali

i. Masukan pengguna: Halo, udah makan belom?

Keluaran:



ii. Masukan pengguna:

5. Analisis Hasil Pengujian

- a. Pada fitur penambahan task baru bot berhasil melakukan penambahan task jika bot menerima masukan tanggal yang valid (berformat DD/MM/YYYY), kode mata kuliah yang sesuai (HHAAAAA dengan H: huruf dan A: angka), jenis tugas yang dikenali, dan topik tugas (ada kata “topik”). Penambahan task berhasil dilakukan walaupun bahasa yang digunakan adalah bahasa sehari-hari. Bot juga menampilkan pesan berhasil menambahkan task ke payar apabila penambahan task berhasil dilakukan.
- b. Pada fitur melihat daftar task yang harus dikerjakan bot berhasil menampilkan semua deadline yang dimiliki (hasil pengujian 4.b.i). Bot juga berhasil menampilkan semua deadline jenis task tertentu (hasil pengujian 4.b.ii), berhasil menampilkan semua deadline hari ini (hasil pengujian 4.b.iii), berhasil menampilkan semua deadline jenis task tertentu hari ini (hasil pengujian 4.b.x), berhasil menampilkan semua deadline antara 2 tanggal masukan (hasil pengujian 4.b.iv), berhasil menampilkan deadline task tertentu di antara 2 tanggal masukan (hasil pengujian 4.b.vi), deadline N hari ke depan (hasil pengujian 4.b.vii), deadline task tertentu N hari ke depan (hasil pengujian 4.b.viii), deadline N minggu ke depan (hasil pengujian 4.b.xii), dan deadline task tertentu N minggu ke depan(hasil pengujian

- 4.b.xi). Masukan untuk melihat daftar deadline ini dapat menggunakan bahasa sehari-hari.
- c. Pada fitur menampilkan deadline pada suatu kode mata kuliah tertentu apabila masukan task benar (ada tersimpan di dalam database) maka akan ditampilkan tanggal deadline dari kode mata kuliah tersebut. Jika mata kuliah yang dimasukkan memiliki lebih dari 1 deadline maka akan ditampilkan tanggal deadline beserta topik dari *task* dari setiap deadline. Jika hanya ada 1 deadline mata kuliah yang dimasukkan maka akan menampilkan tanggal deadline tersebut ke layar. Jika mata kuliah masukan tidak memiliki task yang disimpan di basis data maka bot akan menampilkan pesan khusus. Penambahan deadline ini juga dapat menggunakan bahasa sehari-hari.
 - d. Pada fitur memperbarui task tertentu jika masukan tanggal dan id task benar maka deadline task akan berhasil dilakukan beserta dengan pesan berhasil memperbarui deadline yang ditampilkan ke layar. Apabila id task yang ingin diubah ternyata tidak ada di dalam basis data maka bot akan menampilkan pesan khusus. Fitur memperbarui deadline ini juga dapat menggunakan bahasa sehari-hari.
 - e. Pada fitur untuk menandai bahwa suatu task selesai dikerjakan apabila masukan id task benar maka akan dilakukan penghapusan task di basis data sehingga task tersebut tidak akan muncul kembali di daftar deadline. Selain itu juga akan ditampilkan pesan berhasil menandai bahwa task yang bersangkutan sudah selesai ke layar. Apabila masukan id task tidak benar maka akan diberikan pesan kesalahan oleh bot. Fitur menandai task yang sudah selesai ini juga dapat menerima masukan dari pengguna yang menggunakan bahasa sehari-hari.
 - f. Pada fitur menampilkan opsi help yang difasilitasi oleh bot, bot berhasil menampilkan fitur-fitur yang dimilikinya, daftar kata penting, dan format penulisan untuk tanggal dan kode mata kuliah yang dapat diterima. Pemanggilan menu help ini juga dapat dilakukan dengan menggunakan bahasa sehari-hari.

Berdasarkan uraian di atas dapat kami simpulkan bahwa bot dapat menerima masukan bahasa sehari-hari dan dapat menjalankan semua fitur pada spek dengan baik.

BAB V

Kesimpulan dan Saran

1. Kesimpulan

Pemanfaatan algoritma *String Matching* dan juga *Regular Expression* sangat dibutuhkan dalam pembuatan *chatbot* untuk menganalisis kebutuhan pengguna melalui masukan yang diberikan. Pemakaian *String Matching* ataupun *Regular Expression* dapat disesuaikan dengan kebutuhan pencocokan string. Apabila diperlukan penemuan kata penting pada masukan yang diberikan, akan lebih mudah apabila menggunakan algoritma *String Matching*. Apabila diperlukan pencocokan masukan pengguna dengan *pattern* tertentu, akan lebih mudah dilakukan menggunakan *Regular Expression*. Selain itu, dalam pembuatan aplikasi AntiKeosKeosBot, penerimaan dan pengeluaran data menjadi lebih mudah dengan memanfaatkan basis data yang didukung oleh DBMS yang memadai.

2. Saran

Sebelum membuat aplikasi ini ataupun mempelajarinya, kami menyarankan untuk memahami terlebih dahulu konsep algoritma *String Matching* dan *Regular Expression* untuk mempermudah pembangunan program, demikian juga penggunaan *framework* Flask Python dan DBMS MySQL.

3. Refleksi dan Komentar

Melalui pembuatan program ini, kami mendapatkan pemahaman yang lebih baik mengenai konsep algoritma *String Matching* (khususnya KMP) dan *Regular Expression* dengan menerapkannya langsung pada perancangan dan implementasi program. Penggerjaan tugas besar ini juga menambah pengetahuan kami mengenai pembuatan aplikasi berbasis web lokal dan pemanfaatan basis data menggunakan DBMS MySQL, serta bagaimana menyinkronkan basis data dengan program.

DAFTAR PUSTAKA

- Munir, R. 2021. *Bahan Kuliah IF2211 Strategi Algoritma: Pencocokan String (String/Pattern Matching)*. Diakses pada tanggal 27 April 2021, dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>.
- Khodra, Masayu Leylia. *String Matching dengan Regular Expression*. Diakses pada tanggal 27 April 2021, dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regular-Expression-2019.pdf>.
- Program Studi Teknik Informatika. 2021. *Tugas Besar III IF2211 Strategi Algoritma: Penerapan String Matching dan Regular Expression dalam Pembangunan Deadline Reminder Assistant*. Diakses pada tanggal 27 April 2021, dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Tugas-Besar-3-IF2211-Strategi-Algoritma-2021.pdf>.