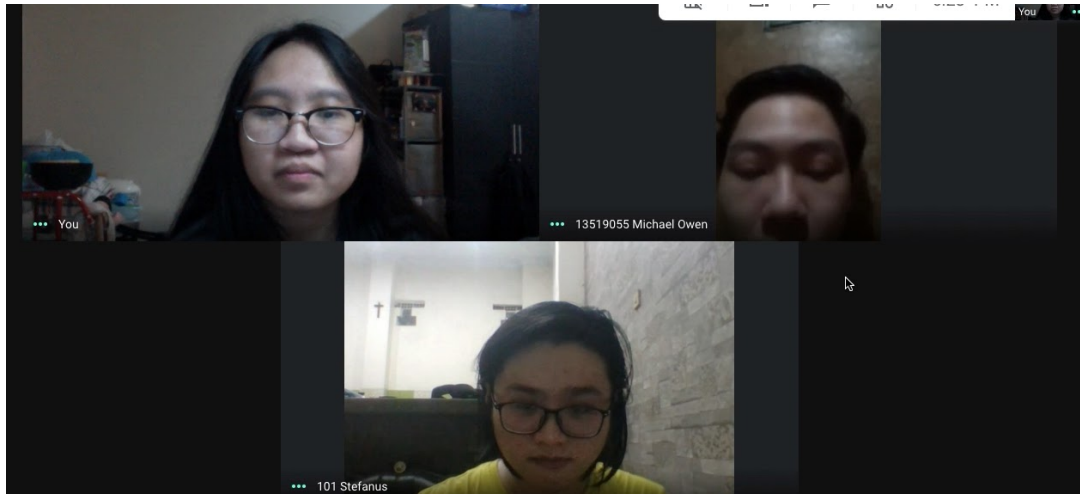


**Tugas Besar 3 IF2211 Strategi Algoritma**  
**Penerapan *String Matching* dan *Regular Expression* dalam**  
**Pembangunan *Deadline Reminder Assistant***  
**Semester II Tahun 2020/2021**



**Kelompok 62**

**CetakCetekKiBOT**

**Michael Owen / 13519055**

**Jeanne D'Arc Amara Hanieka / 13519082**

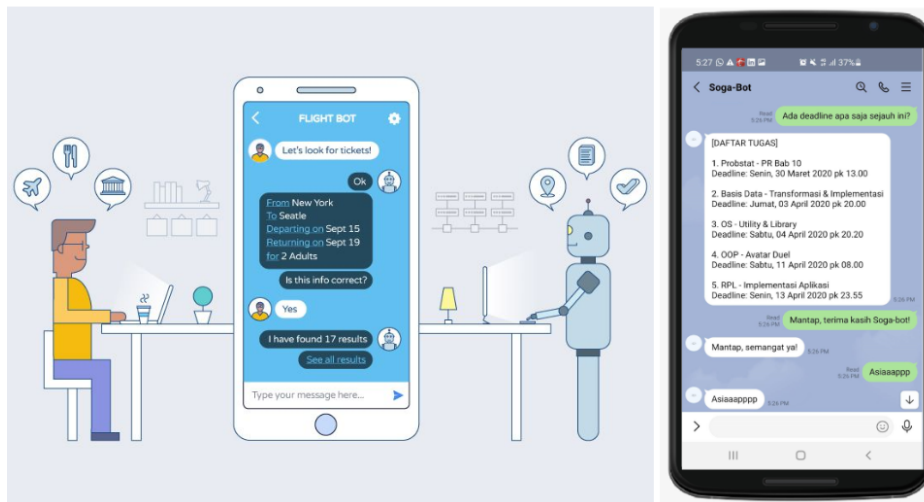
**Stefanus / 13519101**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**JL. GANESHA 10, BANDUNG 40132**

## BAB I

### DESKRIPSI TUGAS

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



Gambar 1.1. Ilustrasi chatbot dan asisten pengingat deadline  
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Tugas-Besar-3-IF2211-Strategi-Algoritma-2021.pdf>)

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 melewati deadline tugas.

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

## BAB II

### LANDASAN TEORI

#### 2.1. Pattern Matching

*Pattern Matching* adalah pencarian *pattern* berukuran  $m$  dari *string* berukuran  $n$  dimana  $n \gg m$  sedemikian rupa sehingga ditemukan lokasi *pattern* di dalam *string* tersebut.

#### 2.2. Skema Umum Algoritma Boyer Moore

Algoritma Boyer Moore memiliki 2 teknik utama yakni the looking glass technique dan the character jump technique. Algoritma ini memiliki beberapa langkah :

1. Algoritma Boyer-Moore mulai mencocokkan *pattern* pada awal teks.
2. Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
  - a. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (mismatch).
  - b. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* dengan memaksimalkan nilai pergeseran good-suffix dan pergeseran bad-character, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks.

#### 2.3. Regular Expression

*Regular Expression* adalah serangkaian karakter yang mendefinisikan sebuah pola pencarian. Pola tersebut biasanya digunakan oleh algoritma pencarian string untuk melakukan operasi cari pada string, atau untuk memeriksa string masukan. Ekspresi reguler merupakan teknik yang dikembangkan dalam bidang ilmu komputer teori dan teori bahasa formal.

Sekarang, ekspresi reguler didukung oleh banyak bahasa pemrograman dan beberapa program lain. Dukungan untuk ekspresi reguler merupakan bagian dari pustaka standar banyak bahasa pemrograman, seperti Java dan Python, dan merupakan bagian dari sintaks bahasa pemrograman lainnya, seperti Perl dan ECMAScript. Implementasi dari

fungsionalitas ekspresi reguler disebut sebuah mesin ekspresi reguler, dan terdapat pustaka-pustaka yang bisa digunakan.

#### **2.4. ChatBot**

Pada Tugas besar kali ini, digunakan bahasa Python sebagai *backend* serta Flask sebagai *frontend*. Pemilihan flask sebagai *frontend* didasarkan kepada ke-*straightforward*-annya yang langsung terhubung dengan *backend* yakni Python.

Selain itu, basis HTML yang merupakan bahasa yang cukup *mainstream* ini bertebaran di daring mempermudah pengimplementasian program apapun dengan bahasa python.

*Chatbot* yang diimplementasikan akan menerima input dari user berupa command apapun yang kemudian akan dicocokkan dengan pattern matching dan regular expression sedemikian rupa sehingga perintah dapat dieksekusi dan menghasilkan keluaran tertentu pada chatbot ini.

## **BAB III**

### **ANALISIS PEMECAHAN MASALAH**

#### **3.1 Langkah-langkah Penyelesaian Masalah setiap Fitur**

Secara umum :

1. Menentukan fungsi apa saja yang bisa dilakukan bot (Menambahkan task baru, melihat daftar task, menampilkan deadline dari suatu task, update tanggal task, menandai suatu task yang telah selesai dikerjakan, menampilkan bantuan, menampilkan pesan error jika input tidak dikenali, memberikan rekomendasi jika ada kesalahan pengetikan)
2. Mengurutkan kondisi kemunculan suatu pattern dari pemanggilan fungsi yang dimaksud dari command, fungsi yang unik misalnya update tanggal task memiliki pattern kata “undur” namun bisa juga mengandung kata kunci dari task lain yang lebih general misalnya dari menambah task yang tidak memiliki pattern khusus sehingga harus diurutkan dari yang paling unik ke yang paling umum
3. Melakukan pencocokan pattern terhadap command dan memanggil fungsi yang paling cocok dengan pattern tersebut

#### **3.2 Fitur fungsional dan arsitektur Chatbot yang dibangun**

##### **1. Menambahkan task baru**

Dalam menambahkan task baru kami melakukan pencocokan string menggunakan fungsi `patternMatching` untuk mencari kecocokan input dengan jenis task yang terdefinisi jika ditemukan maka fungsi `isNewTask` akan mengembalikan nilai `true` dan data jenis task, tanggal deadline, mata kuliah, dan topik task akan di cari menggunakan `regular expression` lalu jika jenis task, tanggal, dan mata kuliah ditemukan (topik opsional) fungsi `tambahTugas` akan dipanggil dan data task baru akan diinsert ke database

##### **2. Melihat daftar task yang harus dikerjakan**

Untuk melihat daftar task akan dilakukan pencocokan string seperti pada fungsi menambahkan tugas baru, pattern yang digunakan yakni kata “apa saja” jika bernilai `true` maka akan ada 2 kondisi yakni menampilkan semua jenis task atau

jenis task tertentu, oleh karena itu kami menggunakan regex untuk mendapatkan pattern jenis task didalam input, jika ditemukan maka daftar yang akan ditampilkan nantinya difilter hanya yang jenisnya sesuai dengan jenis task yang ditemukan pada input. Lalu akan ditentukan rentang waktu jika pattern yang ditemukan adalah x hari kedepan maka akan ditampilkan daftar task dari tanggal saat pencarian dilakukan hingga tanggal pencarian + x, begitu juga dengan hari ini, x minggu kedepan, besok serta berdasarkan input tanggal

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

Untuk menampilkan deadline dari suatu task diperlukan keyword “kapan” dan akan dicari jenis task dan matkul yang ingin ditampilkan deadlinenya menggunakan regex.

### **4. Memperbaharui task tertentu**

Untuk mengupdate tanggal task apabila terjadi pengunduran dari jadwal seharusnya maka dapat menggunakan keyword “undur” yang nanti akan disearch patternnya pada input menggunakan fungsi patternMatching lalu kemudian akan dicari id task dan tanggal barunya menggunakan regex, jika valid maka tanggal dari id task terkait akan diupdate pada database

### **5. Menandai bahwa suatu task sudah selesai dikerjakan**

Untuk menandai suatu task telah selesai dikerjakan maka diperlukan keyword “selesai” atau “beres” dll(dapat dilihat di daftar keyword sinonim yang dapat digunakan) yang akan disearch menggunakan fungsi patternMatching, kemudian akan dicari id task yang ingin diupdate status pengerjaannya jika valid maka status data bersangkutan pada database akan diupdate menjadi TRUE (artinya telah selesai)

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

Untuk menampilkan opsi help maka keyword yang dapat digunakan yakni “bantu” atau “help” yang akan dicocokkan menggunakan patternMatching lalu jika bernilai true opsi help akan ditampilkan oleh bot

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 *Pseudocode Main Program*

**Program CetakCetekKibot**  
{Menerima input query dari pengguna dan memproses query tersebut sehingga menjawab query dari pengguna sesuai dengan harapan}

```
command ← input(querypengguna)
while (command ≠ "Input anda salah")
    reply(querypengguna)
    command ← input(querypengguna)
```

#### 4.2 Struktur Data dan Spesifikasi Program

1. reply(command)  
Fungsi yang memproses masukan dari user dan mencocokkannya dengan fungsi-fungsi yang ada di bawah sehingga bot mengeluarkan jawaban yang diharapkan sesuai dengan input user.
2. DatetoInt(tanggal)  
Mengubah suatu string berformat mm/dd/yyyy menjadi tipe tanggal yang dapat dibaca oleh program
3. bacaDB()  
Membaca file database lalu mereturn list deadline yang ada pada database tersebut.
4. mirip(a,b)  
Menguji kemiripan antara kata a dan kata b serta mengembalikan nilai kemiripan tersebut.
5. karakterUseless(text)  
Menghapus karakter yang tidak dibutuhkan dalam pembacaan query (seperti \, ?, `, \*, dan sebagainya)
6. tambahTugas(tanggal, matkul, jenis, topik)  
Menambahkan tugas baru ke dalam database.
7. tampilTugas()



Menampilkan seluruh tugas yang berstatus belum selesai.

8. tampilTugasDayToDay(hariDua, hariSatu=currdate)

Fungsi untuk menampilkan daftar deadline diantara 2 tanggal dan mengembalikan list deadline.

9. tampilDeadline(jenis, matkul='all')

Berfungsi untuk menampilkan deadline dari matkul dengan jenis yang diinputkan.

10. updateTanggal(id, tanggal)

Berfungsi mengupdate task dengan ID tertentu ke tanggal sesuai parameter.

11. isInDB(tanggal, matkul, jenis)

Mengembalikan boolean true jika ditemukan di database.

12. done(id)

Mengeset status deadline dengan ID sesuai parameter input menjadi true.

13. tampilHelp()

Menampilkan raw text berupa informasi penggunaan bot.

14. patternMatching(pattern, teks)

Menggunakan patternMatching dari Boyer-Moore untuk melihat apakah teks sesuai dengan pattern dan mengembalikan boolean atas jawaban tersebut.

15. isNewTask(input)

Mengecek apakah mengandung keyword untuk menambahkan task baru.

16. isDeadlineList(input)

Mengecek apakah mengandung keyword untuk menampilkan list deadline.

17. isDeadlineTask(input)

Mengecek apakah mengandung keyword untuk menampilkan waktu deadline dari suatu task.

18. isUndurTask(input)

Mengecek apakah mengandung keyword untuk mengundur task.

19. isDoneTask(input)

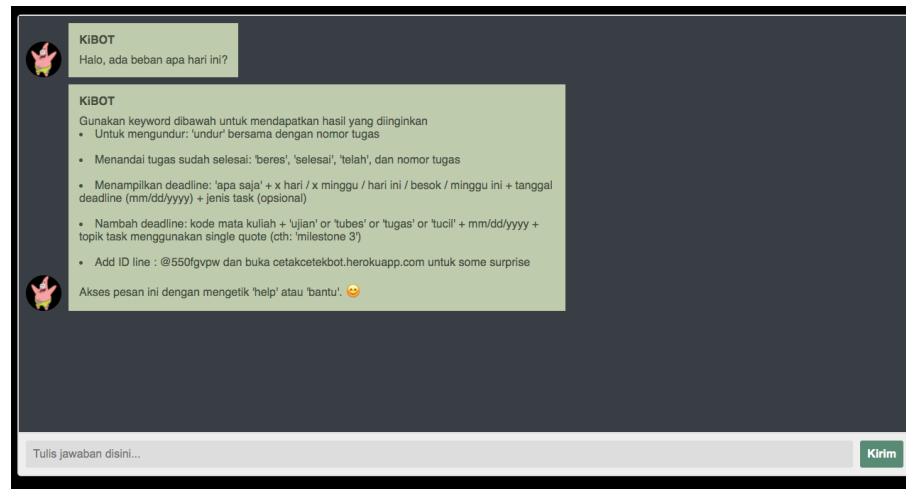
Mengecek apakah mengandung keyword untuk mengubah status task.

20. isHelp(input)

Mengecek apakah mengandung keyword untuk menampilkan bantuan.

### 4.3 Tata Cara Penggunaan Program

Ketika program dijalankan menggunakan perintah `python app.py`, maka akan muncul tab baru pada browser. Pada tab tersebut akan muncul tampilan seperti berikut.



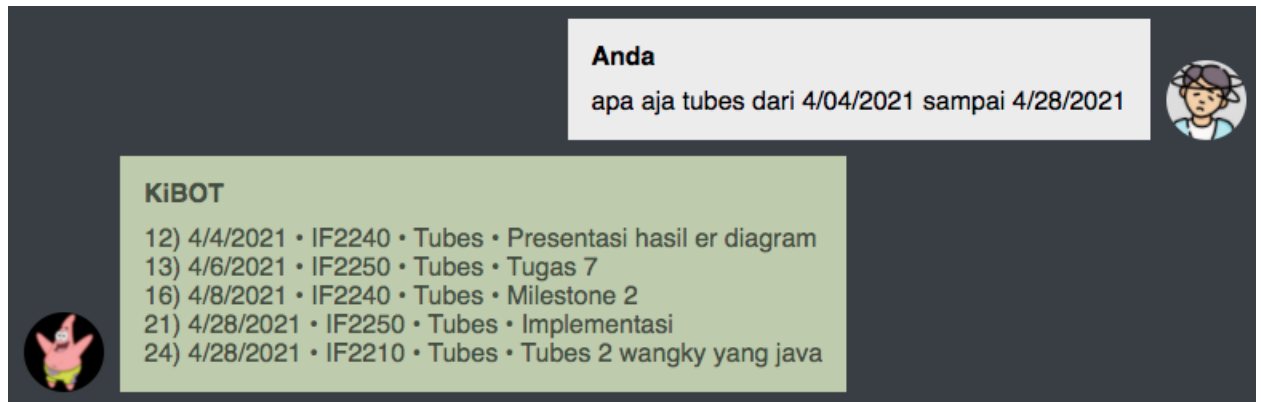
*Gambar 4.3.1 Tampilan awal*

Pengguna dapat memasukkan query pada kolom teks yang ada di bawah. Pengiriman query dapat menggunakan enter atau menekan tombol “kirim”. Pengguna dapat melihat query yang dapat dimasukkan dengan menuliskan ‘help’ atau ‘bantu’.



*Gambar 4.3.2 Tampilan help*

Salah satu contoh penggunaan bot adalah sebagai berikut. Pada tampilan berikut ini, bot menampilkan seluruh tugas besar yang deadline pada tanggal 4 April - 28 April 2021.



*Gambar 4.3.3 Contoh tampilan jawaban BOT*

Bot juga dapat diakses pada [cetakcetekbot.herokuapp.com](https://cetakcetekbot.herokuapp.com) dan LINE @550fgvpw.

#### 4.4 Hasil Pengujian

##### 1. Menambah task baru



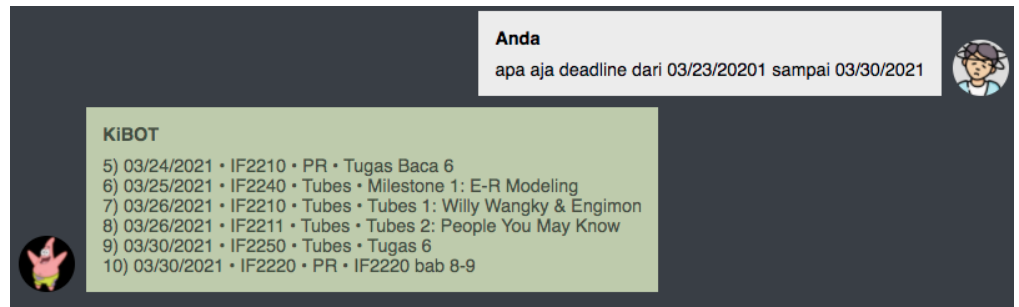
## 2. Melihat daftar task yang harus dikerjakan

### a. Seluruh task yang sudah tercatat oleh assistant

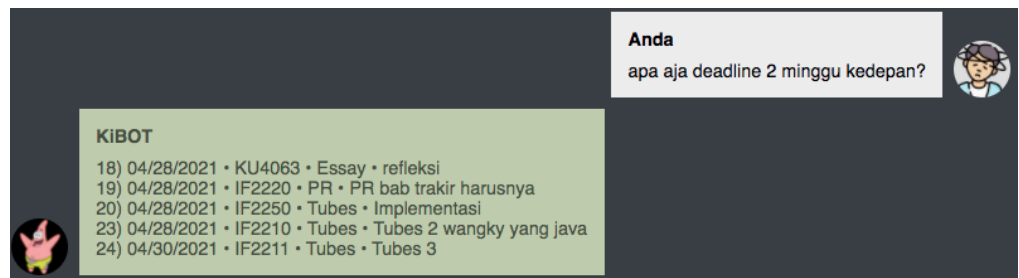


### b. Berdasarkan periode waktu


#### i. Pada periode tertentu (DATE\_1 until DATE\_2)



#### ii. N minggu ke depan




iii. N hari ke depan




**KIBOT**

- 18) 04/28/2021 • KU4063 • Essay • refleksi
- 19) 04/28/2021 • IF2220 • PR • PR bab trakir harusnya
- 20) 04/28/2021 • IF2250 • Tubes • Implementasi
- 23) 04/28/2021 • IF2210 • Tubes • Tubes 2 wangky yang java
- 24) 04/30/2021 • IF2211 • Tubes • Tubes 3


**Anda**  
apa aja deadline 2 minggu kedepan?

iv. Hari ini




**KIBOT**

- 18) 04/28/2021 • KU4063 • Essay • refleksi
- 19) 04/28/2021 • IF2220 • PR • PR bab trakir harusnya
- 20) 04/28/2021 • IF2250 • Tubes • Implementasi
- 23) 04/28/2021 • IF2210 • Tubes • Tubes 2 wangky yang java


**Anda**  
apa aja deadline hari ini

v. Minggu ini




**KIBOT**

- 18) 04/28/2021 • KU4063 • Essay • refleksi
- 19) 04/28/2021 • IF2220 • PR • PR bab trakir harusnya
- 20) 04/28/2021 • IF2250 • Tubes • Implementasi
- 23) 04/28/2021 • IF2210 • Tubes • Tubes 2 wangky yang java
- 24) 04/30/2021 • IF2211 • Tubes • Tubes 3

**Anda**  
apa aja deadline 2 minggu kedepan?


c. Berdasarkan jenis task (kata penting)

i. Sesuai dengan daftar task yang didefinisikan

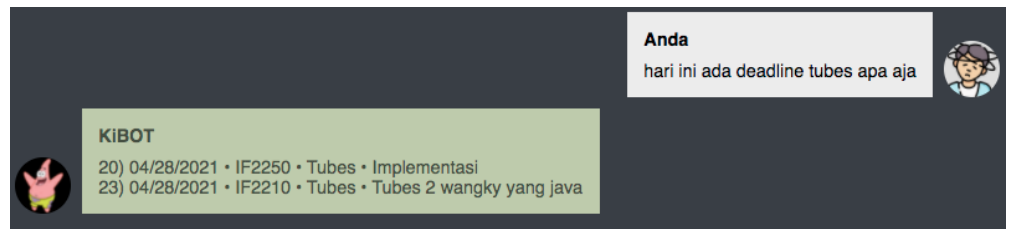


**KIBOT**

- 1) 03/18/2021 • IF2210 • PR • Latihan Responsi Week 9
- 3) 03/23/2021 • IF2240 • PR • IF2240 di stei 19
- 5) 03/24/2021 • IF2210 • PR • Tugas Baca 6
- 10) 03/30/2021 • IF2220 • PR • IF2220 bab 8-9
- 19) 04/28/2021 • IF2220 • PR • PR bab trakir harusnya

**Anda**  
ada PR apa aja

ii. Kombinasi



3. Menampilkan deadline dari suatu task tertentu



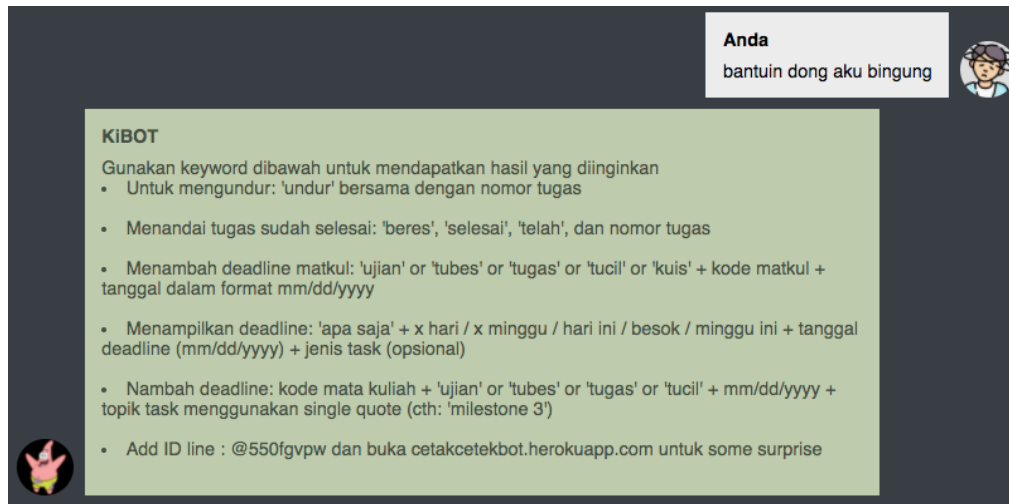
4. Memperbaharui task tertentu



5. Menandai bahwa suatu task sudah selesai dikerjakan



6. Menampilkan opsi help yang difasilitasi oleh assistant



7. Menampilkan pesan error jika assistant tidak dapat mengenali masukan user.



8. Chatbot dapat memberikan rekomendasi kata jika terdapat kesalahan kata (*typo*) pada perintah yang ditulis pengguna.



#### 4.5 Analisis

Bot dapat melakukan semua spesifikasi yang diharapkan dengan baik karena pencocokan string berjalan dengan baik. Selain itu, bot memiliki fungsionalitas yang tepat tujuan serta pengaplikasiannya-pun sesuai dengan *outcome*. Penggunaan kombinasi Regular Expression serta String Matching memberikan kombinasi fleksibilitas dan keketatan yang tepat guna.

## **BAB V**

### **KESIMPULAN, SARAN DAN REFLEKSI**

#### **5.1 Kesimpulan**

String Matching dan Regular Expression dapat dimanfaatkan untuk membuat sebuah ChatBot yang dapat menjadi pengingat tugas. Pada tugas kali ini, telah dibuat sebuah ChatBot beserta tampilan webnya bernama ‘CetakCetekKiBOT’ yang mampu melakukan berbagai fungsi seperti menampilkan *deadline* tugas, menambah *deadline* baru, menandai tugas sudah selesai, dan sebagainya.

#### **5.2 Saran dan Refleksi**

Program ini dapat ditingkatkan dengan memperhatikan beberapa hal, yaitu:

1. Mengatasi *error* apabila input tidak sesuai. Terdapat banyak variasi input yang dapat dimasukkan oleh pengguna, sehingga akan menjadi lebih baik apabila dilakukan penanganan terhadap berbagai macam input tersebut.
2. Lebih memanfaatkan penggunaan database agar bisa digunakan oleh beberapa user

Program ini masih jauh dari kata sempurna, namun dapat melakukan fungsi-fungsi dasar yang diharapkan dari program. Tugas ini membantu kami untuk memahami String Matching dan Regular Expression dengan lebih baik dan mampu membuat kami membayangkan sistem seperti apa yang digunakan pada ChatBot-ChatBot yang tersedia di pasaran.

#### **5.3 Bonus**

Kamipun mengimplementasikan Tugas Besar 3 Strategi Algoritma ini ke website `cetakcetekbot.herokuapp.com` serta OA Line di `@550fgvpw`. Namun perlu digarisbawahi bahwasanya *deploy* di website dan OA Line ini menggunakan 1 server yaitu heroku, sehingga kecepatan dari tiap command yang diberikan acapkali membutuhkan waktu lebih.



## REFERENSI

Levitin, Anany. 2003. *Introduction to the Design & Analysis of Algorithms*. Boston: Addison-Wesley

“String Matching dengan Regular Expression” oleh Masayu Leylia Khodra (2019)  
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>