

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

LAPORAN TUGAS BESAR 3

Diajukan sebagai salah satu Tugas Besar 3

IF2211 Strategi Algoritma Semester II tahun 2020/2021

Oleh:

13519023 - Ilyasa Salafi Putra Jamal

13519051 - Yudi Alfayat

13519172 - Muhammad Zubair



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2021**

DAFTAR ISI

DAFTAR ISI

BAB I

DESKRIPSI TUGAS

[1.1 Latar belakang:](#)

[1.2 Deskripsi Tugas:](#)

[1.3 Fitur-Fitur Aplikasi:](#)

BAB II

LANDASAN TEORI

[2.1 String Matching](#)

[2.1.2 The Knuth-Morris-Pratt \(KMP\) Algorithm](#)

[2.1.3 The Boyer-Moore \(BM\) Algorithm](#)

[2.1.4 Regular Expression](#)

[2.2 Chatbot](#)

BAB III

ANALISIS PEMECAHAN MASALAH

[3.1 Langkah Penyelesaian](#)

[3.2 Fitur Fungsional dan Arsitektur yang Dibangun](#)

BAB IV

IMPLEMENTASI DAN PENGUJIAN

[4.1 Spesifikasi Teknis Program](#)

[4.2 Tata Cara Penggunaan Program](#)

[4.3 Hasil Pengujian](#)

[4.4 Analisis Hasil Pengujian](#)

BAB V

KESIMPULAN, SARAN, REFLEKSI, DAN KOMENTAR

[5.1 Kesimpulan](#)

[5.2 Saran, Refleksi, dan Komentar](#)

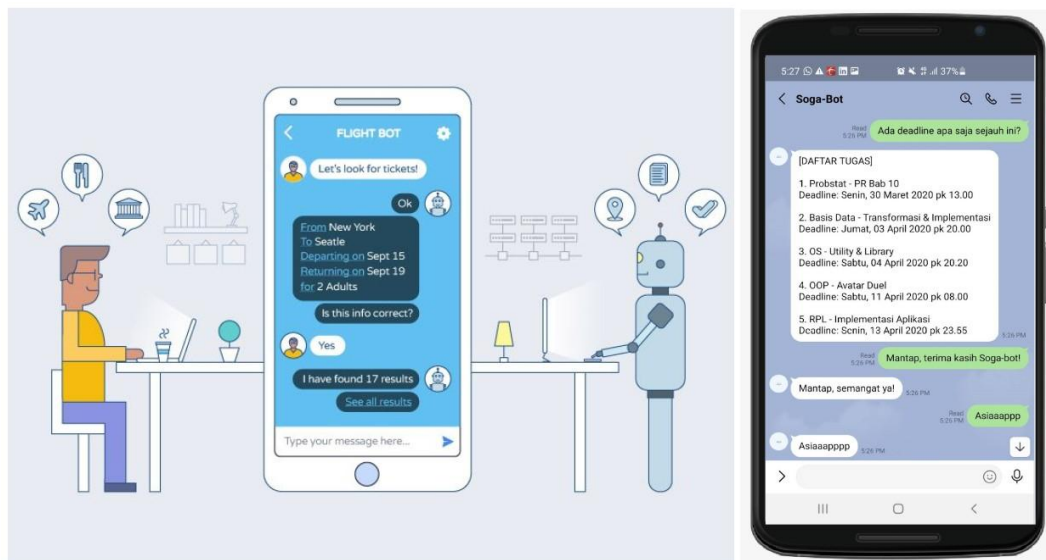
DAFTAR PUSTAKA

BAB I

DESKRIPSI TUGAS

1.1 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 melewatkan deadline tugas.

1.2 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.

1.3 Fitur-Fitur Aplikasi:

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

1. Menambahkan task baru

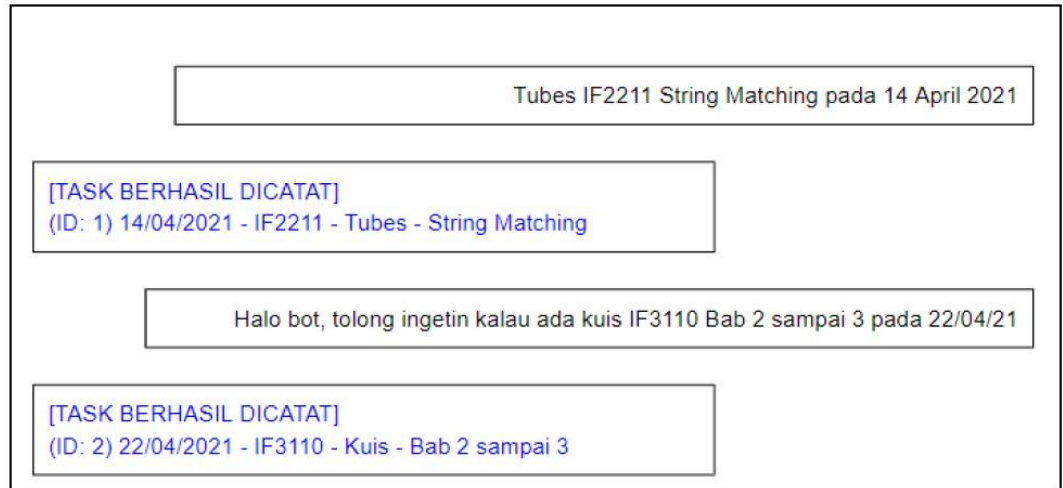
- a. Suatu kalimat diklasifikasikan sebagai suatu task apabila mengandung semua komponen berikut ini:
 - Tanggal (format dibebaskan)
 - Kode Mata Kuliah / Nama Mata Kuliah (dibebaskan)
 - Jenis Tugas (berdasarkan daftar kata penting yang sudah disediakan)
 - Topik Tugas (tidak ada batasan)
- b. Point 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:

Apa saja deadline yang dimiliki sejauh ini?

[Daftar Deadline]
1. (ID: 1) 14/04/2021 - IF2211 - Tubes - String Matching
2. (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

Apa saja deadline antara 03/04/2021 sampai 15/04/2021?

[Daftar Deadline]
1. (ID: 2) 14/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

Deadline 3 minggu ke depan apa saja?

[Daftar Deadline]
1. (ID: 1) 14/04/2021 - IF2211 - Tubes - String Matching
2. (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

Deadline 1 hari ke depan apa saja?

Tidak ada

3 minggu ke depan ada kuis apa saja?

[Daftar Deadline]
1. (ID: 2) 22/04/2021 - IF3110 - Kuis - Bab 2 sampai 3

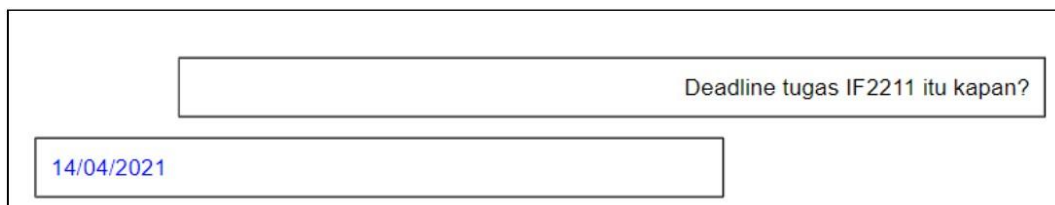
Keterangan penting:

- 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



The diagram illustrates a chatbot interaction within a rectangular frame. At the top, a text box contains the user's query: "Deadline tugas IF2211 itu kapan?". Below this, a response box displays the date "14/04/2021" in blue text.

4. Memperbaharui task tertentu

- Memperbarui tanggal dari suatu task (dalam kehidupan nyata, tentu ada kejadian dimana deadline dari suatu task diundur)
- Perintah yang dimasukkan meliputi 1 keyword untuk memperbaharui suatu task dan nomor task tertentu.
- Misalnya:
 - “Deadline task X diundur menjadi 28/04/2021” dimana X merupakan nomor ID dari suatu task.
- Apabila task berhasil diperbaharui, Chatbot akan menampilkan pesan sukses memperbaharui suatu task. Sebaliknya, Chatbot akan menampilkan pesan error apabila task yang dimaksud tidak dikenali oleh Chatbot (belum masuk ke dalam Daftar Task)

5. Menandai bahwa suatu task sudah selesai dikerjakan

- 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.
- 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

- Berisikan command-command yang dapat digunakan oleh user
- Misalnya: “Apa yang bisa assistant lakukan?”
- Bot akan memberikan hasil berupa daftar kata-kata yang bisa digunakan untuk menambahkan dan melihat daftar task (setiap kelompok bebas membentuknya seperti apa)
- Contoh interaksi

Apa yang bisa assistant lakukan?

[Fitur]

1. Menambahkan task baru
2. Melihat daftar task
3.
4. *(dan seterusnya)*

[Daftar kata penting]

1. Kuis
2. Ujian
3. Tupil
4. Tubes
5.

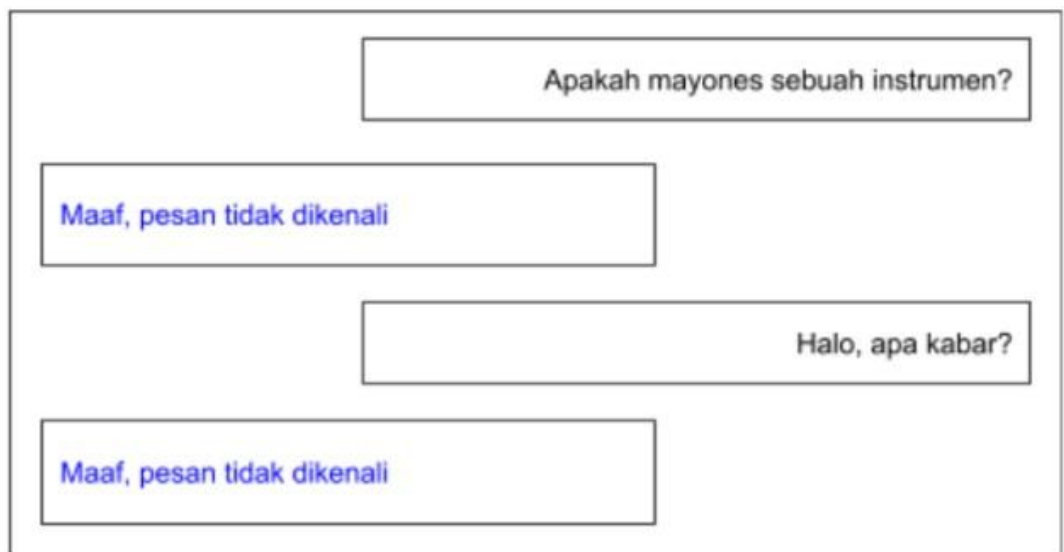
7. Mendefinisikan list kata penting terkait apakah itu merupakan suatu task atau tidak

- Minimal terdapat 5 kata penting berbeda, contohnya adalah: [“Kuis”, “Ujian”, “Tupil”, “Tubes”, “Praktikum”]
- Kata penting akan digunakan pada penentuan jenis tugas dari suatu task.

- c. Daftar kata penting tidak perlu dibuat dinamis, cukup static saja atau hardcoded

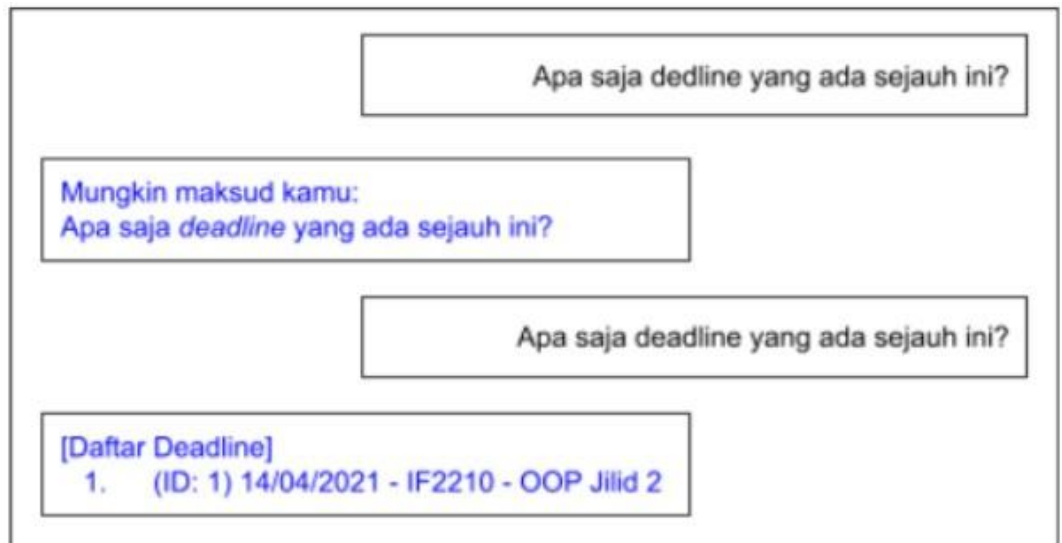
8. Menampilkan pesan error jika assistant tidak dapat mengenali masukan user.

- a. Masukan yang tidak termasuk ke dalam jenis pesan di poin 1 sampai 4 dapat dikategorikan sebagai masukan tak dikenali.
- b. Error message dibebaskan sesuai kreativitas mahasiswa
- c. Contoh interaksi



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

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



- c. Ada berbagai matriks yang dapat dimanfaatkan untuk mencari kemiripan kata, salah satunya adalah Levenshtein distance yang diukur melalui pendekatan dynamic programming.

BAB II

LANDASAN TEORI

2.1 String Matching

String matching atau pattern matching dapat didefinisikan sebagai berikut:

Diberikan:

1. T: teks(text), yaitu (long) string yang panjangnya n karakter
2. P: pattern, yaitu string dengan panjang m karakter (asumsi $m \ll n$) yang akan dicari di dalam teks.

String matching atau pattern matching berarti pencarian (find atau locate) lokasi pertama di dalam teks yang bersesuaian dengan pattern.

Jika suatu string memiliki panjang m sebagaimana berikut :

$$S = X_0 X_1 \dots X_{m-1}$$

Maka yang disebut dengan Prefix adalah substring dari S dari bagian awal, yaitu $S[0..k]$ dengan $0 \leq k \leq m-1$, sedangkan yang disebut dengan Suffix adalah substring dari bagian akhir, yaitu $S[k..m-1]$.

String matching secara umum dapat dibagi menjadi dua, yaitu Exact Matching dan Approximate/Regex Matching. Terdapat beberapa metode yang dapat digunakan untuk melakukan exact matching, seperti pendekatan naive/brute force, DFA, KMP, dan BM. Algoritma yang digunakan dalam tugas kali ini adalah algoritma KMP, BM, dan juga Regex Matching.

Pada algoritma naive/brute force, setiap huruf pada text akan diperiksa dengan setiap huruf pada pattern. Pattern akan selalu bergerak satu karakter tiap pengecekan melalui text. Ini menyebabkan algoritma ini sangat buruk terutama jika terdapat banyak karakter matching sebelum menemukan ketidakcocokan, seperti contoh dibawah ini:

T: AAAAAAAAAAAB

P: AAAB

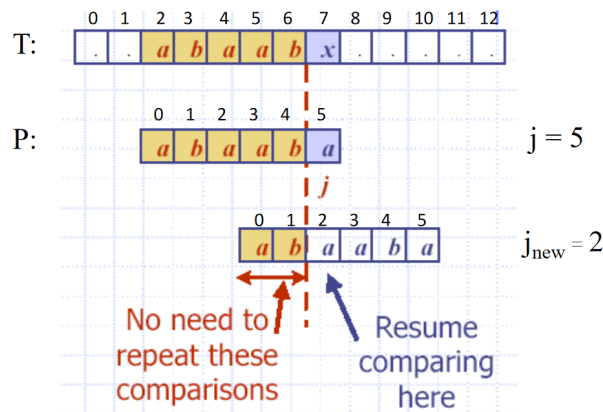
Jika menghadapi kasus seperti diatas, pencocokan dapat terjadi sebanyak $m(n-m+1)$, yang menghasilkan kompleksitas algoritma $O(mn)$.

2.1.2 The Knuth-Morris-Pratt (KMP) Algorithm

The Knuth-Morris-Pratt (KMP) adalah algoritma yang mencari pola/pattern dalam suatu text dengan order kiri ke kanan. Algoritma KMP ini ditemukan oleh Donald E. Knuth. Algoritma ini adalah jenis Exact String Matching Algorithm yang merupakan pencocokan string secara tepat dengan susunan karakter dalam string

yang dicocokkan memiliki urutan dan jumlah karakter yang sama. Algoritma The Knuth-Morris-Pratt (KMP) lebih pintar dalam menggeser-geser pola dibandingkan algoritma brute force.

Pada algoritma KMP, jika terjadi ketidakcocokan antara pattern dengan text di indeks ke- j pattern ($T[i] \neq P[j]$), pattern akan digeser sebesar prefix terbesar $P[0..j-1]$ yang juga merupakan suffix $P[1..j-1]$. Berikut ini ilustrasinya:



Gambar 2. Ilustrasi Algoritma KMP

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>

KMP melakukan preprocessing terhadap pattern untuk menemukan kecocokan prefix dengan pattern itu sendiri. Misalnya :

j = mismatch position in $P[]$

k = position before the mismatch ($k = j - 1$).

Fungsi pinggiran KMP $b(k)$ didefinisikan sebagai ukuran prefix terbesar dari $P[0..k]$ yang juga merupakan suffix $P[1..k]$. Fungsi pinggiran $b(k)$ biasanya direpresentasikan dalam bentuk tabel seperti contoh dibawah ini:

$(k = j-1)$

P: abaaba
j: 012345

j	0	1	2	3	4	5
$P[j]$	a	b	a	a	b	a
k	-	0	1	2	3	4
$b(k)$	-	0	0	1	1	2

$b(k)$ is the size of the largest border.

Gambar 3. Ilustrasi Representasi Fungsi Pinggiran $b(k)$

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>

Algoritma KMP memiliki kompleksitas waktu $O(m+n)$, yaitu $O(m)$ dari menghitung fungsi pinggiran, dan $O(n)$ dari pencarian string. Sehingga algoritma ini jauh lebih efisien dibandingkan algoritma brute force.

2.1.3 The Boyer-Moore (BM) Algorithm

Algoritma Boyer-Moore merupakan algoritma pencocokan string persis (exact string matching). Algoritma ini dianggap sebagai algoritma paling efisien untuk melakukan pencocokan string. Algoritma yang ditemukan oleh Bob Boyer dan J. Strother Moore. Algoritma ini sering digunakan pada teks editor yang digunakan untuk perintah search atau commands. Berbeda dengan algoritma lainnya, algoritma ini mencocokkan pattern dimulai dari karakter paling kanan.

Pencocokan string dengan menggunakan Algoritma Boyer-Moore terdiri atas dua teknik.

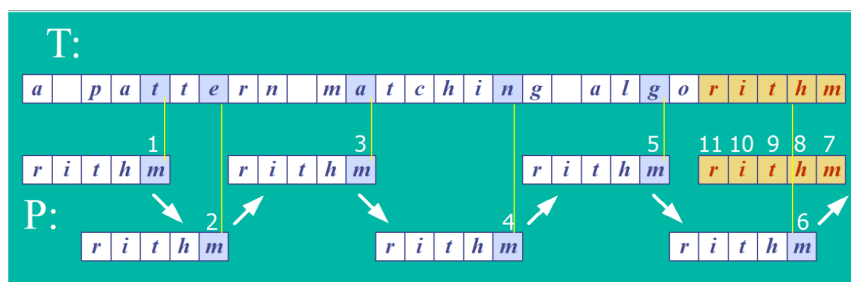
1. The Looking-Glass Technique

The Looking-Glass Technique melakukan perbandingan suatu karakter akhir pada kata w dengan suatu karakter pada teks s . Jika karakter tersebut sama maka jendela karakter akan berjalan mundur pada kedua string dan mengecek kembali kedua karakter. Mencari Suatu kecocokan String pada Teks dengan pola yang akan dicari dengan cara memindahkan atau menggesernya sampai Teks string selesai.

2. The Character-Jump Technique

Character-jump Technique melakukan suatu aksi ketika perbandingan antara dua karakter yang berbeda. Ada dua aksi yang tergantung pada teks s dan kata w yang dimiliki; jika p yaitu karakter pada s yang sedang diproses yang tidak cocok maka ada dua kemungkinan aksi. Mencari karakter yang sesuai dan cara penggeseran sebuah karakter perbandingan terakhir.

Kedua teknik di atas digunakan bersama-sama dalam algoritma boyer moore. Berikut ini ilustrasi contoh penggunaan algoritma ini:



Gambar 2. Ilustrasi Algoritma BM

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>

2.1.4 Regular Expression

Regular Expression atau Regex merupakan sebuah teks (string) yang mendefinisikan sebuah pola pencarian sehingga dapat membantu kita untuk melakukan *matching* (pencocokan), *locate* (pencarian), dan manipulasi teks.

Konsep tentang regex pertama kali muncul di tahun 1951, ketika seorang ilmuwan matematika bernama Stephen Cole Kleene memformulasikan definisi tentang *bahasa formal*.

Regular Expression sering digunakan pada search engine, dialog of processors dan text editors seperti SED dan AWK dan analisis lexical. Sudah banyak bahasa pemrograman yang menyediakan regular expression, baik secara bawaan maupun dengan menggunakan library.

2.2 Chatbot

Chatbot adalah aplikasi perangkat lunak yang digunakan untuk melakukan percakapan melalui teks atau text-to-speech. Desain chatbot dibuat sedemikian rupa sehingga memiliki perilaku yang menyerupai lawan bicara manusia. Chatbot banyak digunakan untuk berbagai kebutuhan, seperti customer service, request routing, personal assistant, event reminder, atau information gathering.

Pembuatan dan pengembangan chatbot adalah bidang yang masih memiliki banyak ruang untuk berkembang. Perkembangan chatbot dapat dikaitkan secara erat dengan perkembangan artificial intelligence dan machine learning. Pengembangan chatbot terbukti menjadi bidang yang sulit dan chatbot yang banyak dikembangkan sekarang masih memiliki beberapa batasan, beberapa batasan tersebut antara lain:

1. Database untuk menghasilkan output terbatas, sehingga chatbot dapat gagal merespon terhadap query yang tidak terdaftar di database.
2. Performa chatbot bergantung pada pemrosesan bahasa, sehingga kemampuan chatbot dapat terganggu oleh beberapa hal seperti typo, bahasa informal/gaul, kesalahan grammar, atau istilah lokal.
3. Chatbot masih belum dapat menangani banyak percakapan sekaligus, hanya dapat merespon terhadap satu percakapan dalam satu waktu.
4. Chatbot memerlukan database yang besar untuk dapat berkomunikasi secara efektif dan natural.

BAB III

ANALISIS PEMECAHAN MASALAH

3.1 Langkah Penyelesaian

Pada tugas besar ini, kami diminta untuk membuat aplikasi chatbot dengan kegunaan sebagai reminder deadline. Aplikasi ini dibuat berbasis web dan dengan memanfaatkan algoritma boyer-moore (BM) atau algoritma knuth-morris-pratt (KMP) dan regular expression untuk mengenali chat dari pengguna, serta menyimpan data deadline dalam database supaya data dapat diminta kembali oleh pengguna.

Setelah mempertimbangkan perbandingan antara algoritma KMP dan BM untuk exact string matching, kami memutuskan untuk menggunakan algoritma BM karena menurut kami, untuk kasus membuat chatbot, akan menggunakan alphabet yang cukup besar sehingga algoritma BM lebih cocok karena ukuran alphabet tersebut.

Kami merancang aplikasi untuk menerima input chat dari pengguna. Chat dapat berupa percakapan natural seperti “Bot, ingatkan tubes IF2211 topik string matching deadlinenya 28/04/2021”. Aplikasi kemudian akan memproses query chat dari pengguna untuk melakukan aksi yang tepat. Pemrosesan query chat tersebut dilakukan melalui string matching menggunakan algoritma boyer-moore dan regex terhadap database kata-kata penting. Berdasarkan kata apa saja yang cocok, aplikasi kemudian melakukan aksi yang sesuai, seperti menambah tugas, menampilkan deadline, menampilkan fitur, menandai tugas telah selesai, dan sebagainya.

3.2 Fitur Fungsional dan Arsitektur yang Dibangun

Aplikasi chatbot yang kami buat ini dikembangkan menggunakan bahasa Javascript, salah satu bahasa yang mendukung penggunaan regular expression. Aplikasi ini juga dikembangkan sebagai aplikasi web, dengan penyimpanan database menggunakan Firebase.

Fitur-fitur yang dimiliki aplikasi chatbot yang kami kembangkan adalah:

1. Menambahkan tugas baru
2. Melihat daftar seluruh tugas
3. Melihat daftar deadline tugas pada tanggal tertentu
4. Menampilkan deadline suatu tugas tertentu
5. Memperbaharui deadline tugas tertentu
6. Menandai suatu tugas telah selesai
7. Menampilkan opsi *help* yang difasilitasi chatbot

8. Menampilkan daftar kata-kata penting

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi Teknis Program

Struktur data yang digunakan :

1. Array listKataPenting
Menyimpan daftar kata penting.
2. Array listTugas
Menyimpan daftar tugas.
3. Array listUbahTask
Menyimpan daftar kata yang merujuk ke fitur ubah deadline tugas.
4. Array listTandai
Menyimpan daftar kata yang merujuk ke fitur tandai tugas telah selesai.
5. Array listMelihatDaftarTask
Menyimpan daftar kata yang merujuk ke fitur menampilkan semua tugas.
6. Array listMenampilkanDeadline
Menyimpan daftar kata yang merujuk ke fitur menampilkan deadline tugas tertentu.
7. Array listHelp
Menyimpan daftar kata yang merujuk ke fitur *help*.
8. Array listWaktu
Menyimpan daftar kata yang merujuk ke waktu deadline.
9. Array listWaktuTambahTask
Menyimpan penanda waktu menambah tugas.
10. Dictionary dictTask
Menyimpan daftar tugas dengan id tugas sebagai key-nya.
11. Dictionary dictTaskSelesai
Menyimpan daftar tugas yang telah selesai dengan id tugas sebagai key-nya.

Fungsi dan Prosedur yang dibuat :

1. boyerMoore(string, pattern)

Algoritma boyer-moore untuk pattern matching, menggunakan shiftDict(pattern) sebagai fungsi batas. Menerima argumen string, yaitu text, dan pattern.

2. shiftDict(pattern)

Digunakan sebagai fungsi batas algoritma boyer-moore. Menerima argumen pattern

3. isUpdate(message)

Mengembalikan true jika fitur yang diminta adalah memperbaharui deadline tugas. Menerima argumen message, yaitu chat dari pengguna. Menggunakan algoritma boyer moore.

4. isTandai(message)

Mengembalikan true jika fitur yang diminta adalah menandai tugas telah selesai. Menerima argumen message, yaitu chat dari pengguna. Menggunakan algoritma boyer moore.

5. isMenampilkanDeadline(message)

Mengembalikan true jika fitur yang diminta adalah menampilkan deadline suatu tugas. Menerima argumen message, yaitu chat dari pengguna. Menggunakan algoritma boyer moore.

6. isMelihatDaftarTask(message)

Mengembalikan true jika fitur yang diminta adalah menampilkan daftar seluruh tugas. Menerima argumen message, yaitu chat dari pengguna. Menggunakan algoritma boyer moore.

7. isHelp(message)

Mengembalikan true jika fitur yang diminta adalah menampilkan *help* dari chatbot. Menerima argumen message, yaitu chat dari pengguna. Menggunakan algoritma boyer moore.

8. isExistKataWaktu(message)

Mengembalikan true jika terdapat kata yang merujuk ke waktu deadline di message dari pengguna. Menggunakan algoritma boyer moore.

9. decision(message)

Mengembalikan jenis fitur yang diminta pengguna, mengembalikan pesan error jika message dari pengguna tidak dikenali, dan mencatat tugas baru jika fitur yang diminta adalah mencatat tugas baru. Menggunakan fungsi-fungsi diatas.

10. `getJenisTugas(string, listJenisTugas)`

Mengembalikan jenis tugas yang dimaksud dari string, yaitu text dari pengguna, berdasarkan `listJenisTugas` yang berdasarkan kata penting. Menggunakan algoritma boyer moore.

11. `getTanggal(string)`

Mengembalikan tanggal deadline dari string, yaitu text dari pengguna. Menggunakan regular expression.

12. `getMatkul(string)`

Mengembalikan kode mata kuliah informatika dari string, yaitu text dari pengguna. Menggunakan regular expression.

13. `getTopic(message)`

Mendapatkan topik atau materi yang berhubungan dengan tugas dari message pengguna. Menggunakan regular expression.

14. `getHelpMessage()`

Menampilkan pesan dari fitur *help*.

15. `selfChat()`

Program utama (main), menjalankan aplikasi chatbot. Menggunakan `decision(message)`, `sendChat(user, message)`, dan `friendChat(messageResponse)`.

16. `friendChat(message)`

Memberikan respon chat pengguna. Menggunakan `sendChat(user, message)`.

17. `sendChat(user, message)`

Mengirimkan chat, baik dari pengguna, maupun dari chatbot. Menggunakan `insertChatToLogs(chatbox)`, `getChatBox(use, message)`, dan `escapeHtml(str)`.

18. `insertChatToLogs(chatbox)`

Memasukkan chat ke dalam penyimpanan chatlogs. Menggunakan `autoScroll()`.

19. `getChatBox(use, message)`

Memberikan penampilan dan visualisasi chatbox pada aplikasi.

20. `escapeHtml(str)`

Mendapatkan chat dari user. Menggunakan regular expression.

21. `autoScroll()`

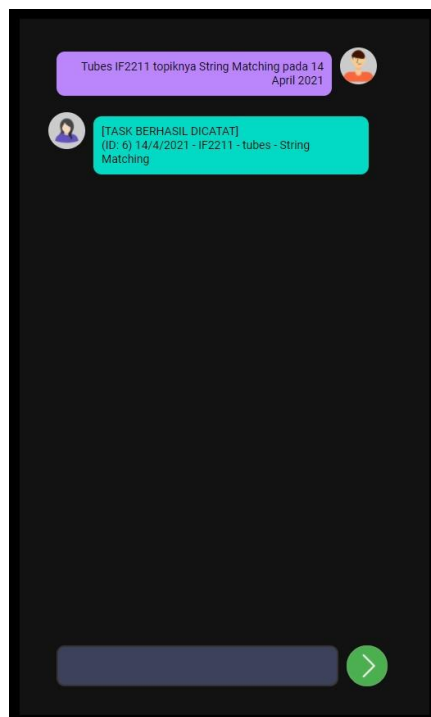
Menggeser layar ke bawah saat chat sudah mencapai puncak layar.

4.2 Tata Cara Penggunaan Program

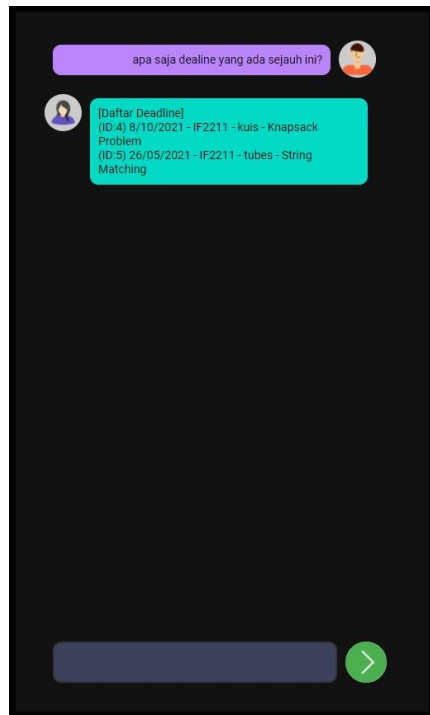
- Masuk ke folder src, lalu buka file `index.html`
- Atau melalui link <https://chatbot-remainder-assist-f2d5d.web.app>
- Untuk memanggil fitur, cukup chat ke chatbot dengan bahasa natural
- Ketika menambahkan tugas, untuk mendefinisikan topiknya, harus menggunakan kata “topik”
- Untuk mendefinisikan tugas dengan id tertentu, setelah kata “task” perlu diikuti dengan id task tersebut

4.3 Hasil Pengujian

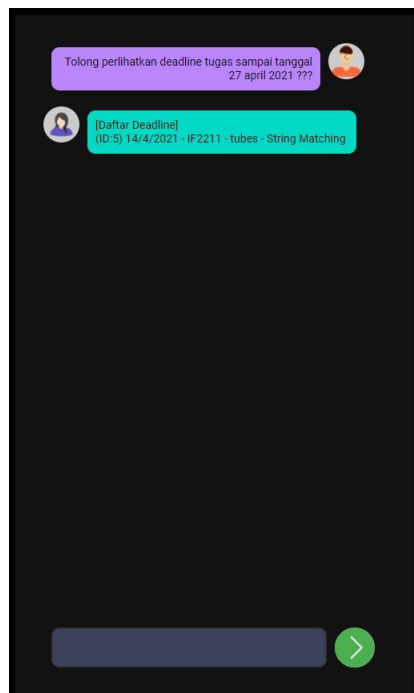
1. Menambahkan tugas baru

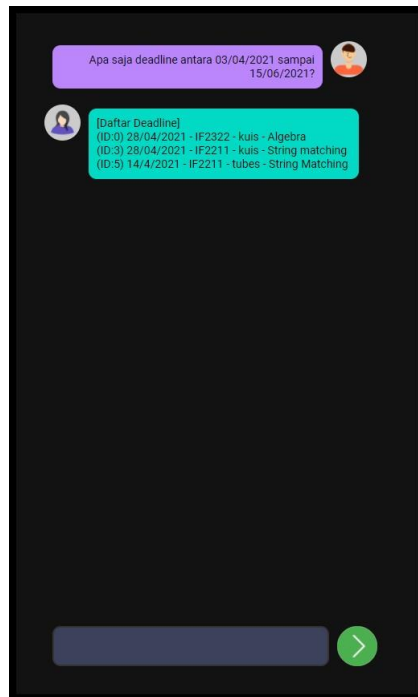


2. Melihat daftar seluruh tugas



3. Melihat daftar deadline tugas pada tanggal tertentu

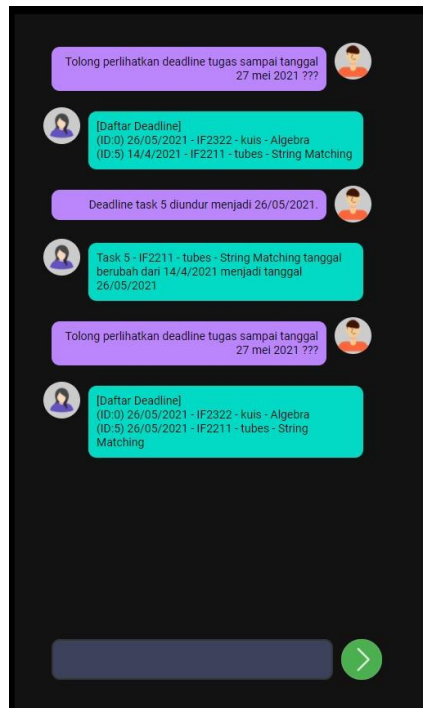




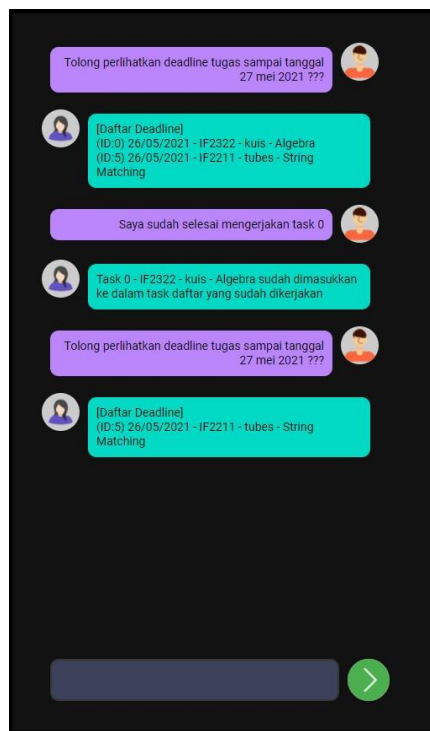
4. Menampilkan deadline suatu tugas tertentu



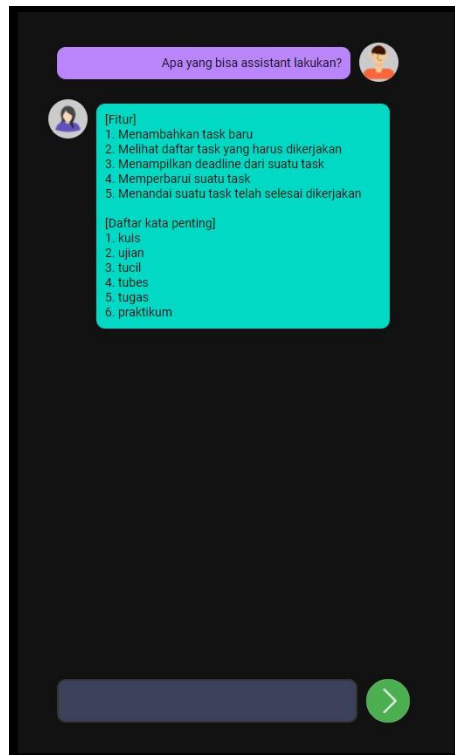
5. Memperbaharui deadline tugas tertentu



6. Menandai suatu tugas telah selesai



7. Menampilkan opsi *help* yang difasilitasi chatbot dan menampilkan kata-kata penting



4.4 Analisis Hasil Pengujian

Aplikasi yang telah kami implementasikan telah lolos dari pengujian yang kami lakukan diatas. Walaupun memang terdapat beberapa batasan dari aplikasi kami, seperti kata kunci tertentu yang harus ada supaya perintah dikenali, tetapi menurut kami, aplikasi ini sudah cukup dekat dalam mengenali perintah dengan bahasa natural.

Sementara itu dari segi fungsional tanpa melihat segi pengenalan bahasa, aplikasi kami telah dapat melakukan fitur-fitur wajib yang diminta. Kami juga telah melakukan pengecekan terhadap penyimpanan database aplikasi ini dan dapat mengatakan bahwa data tersimpan dan diolah dengan baik.

Dari segi penampilan, menurut kami penampilan yang diberikan aplikasi kamu sudah menarik. UI tidak diisi dengan banyak clutter dan tampilan minimalistiknya serta latar belakangnya yang gelap membuatnya ringan untuk dipandang.

BAB V

KESIMPULAN, SARAN, REFLEKSI, DAN KOMENTAR

5.1 Kesimpulan

- Pada akhir tugas besar ini, telah dibuat sebuah program aplikasi chatbot untuk pencatat dan pengingat deadline tugas.
- Program ini dibuat berbasis web, dengan bahasa Javascript, dan menggunakan sistem database Firebase.
- Program ini memanfaatkan algoritma boyer-moore untuk Exact String Matching dan juga Regex String Matching untuk mengenali perintah dan mengekstrak informasi dari chat pengguna.
- Menurut kami, untuk kasus membuat chatbot, akan menggunakan alphabet yang cukup besar sehingga algoritma BM lebih cocok karena ukuran alphabet tersebut.

5.2 Saran, Refleksi, dan Komentar

Tugas besar yang satu ini kami rasa memiliki topik yang cukup menarik. Kami sudah pernah memiliki pengalaman menggunakan chatbot sebelumnya, sehingga mempelajari dan langsung menerapkan konsep-konsep yang ada dibalik chatbot tersebut sangatlah menarik dan dekat dengan kehidupan. Kami juga baru menyadari seberapa kompleks bahasa manusia jika harus didekomposisi dan dipelajari. Selain itu, kami juga mempelajari pengembangan web dan database yang menurut kami akan sangat berguna di masa yang akan datang.

Sayangnya terdapat beberapa hal yang menyebabkan kami kurang dapat menikmati tugas yang satu ini. Beberapa diantaranya adalah waktu pemberian tugas deadline-nya yang sangat berdekatan dengan beberapa tugas besar lainnya, sehingga waktu dan usaha kami sudah sangat terkuras mengerjakan tugas besar lainnya. Hal lain yang cukup mengganggu adalah penggunaan regex dalam bahasa Javascript yang agak sulit dipahami, dan agak berbeda dengan penggunaannya dalam bahasa Python.

Walaupun begitu, kami cukup puas dengan tugas besar ini karena kami telah mempelajari beberapa hal baru, seperti bahasa Javascript, pengembangan web, dan penggunaan regex. Jika kami perlu memberi saran, saran dari kami adalah untuk menjauhkan deadline tugas besar ini dari tugas besar lainnya.

DAFTAR PUSTAKA

1. <https://nero.trunojoyo.ac.id/index.php/nero/article/download/18/16>
2. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>
3. <https://en.wikipedia.org/wiki/Chatbot>