

# LAPORAN TUGAS BESAR

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

Dibuat dalam rangka:  
Tugas Besar 3 IF-2211 Strategi Algoritma

Oleh:

KANG VAPOR	KANG CUPU PGN JAGO	KANG OSU
Nizamixavier Rafif Lutvie	Wisnu Aditya Samiadji	Tanur Rizaldi Rahardjo
13519085	135191093	13519214

**“KangBOT”**



**PROGRAM STUDI TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2020**

# DAFTAR ISI

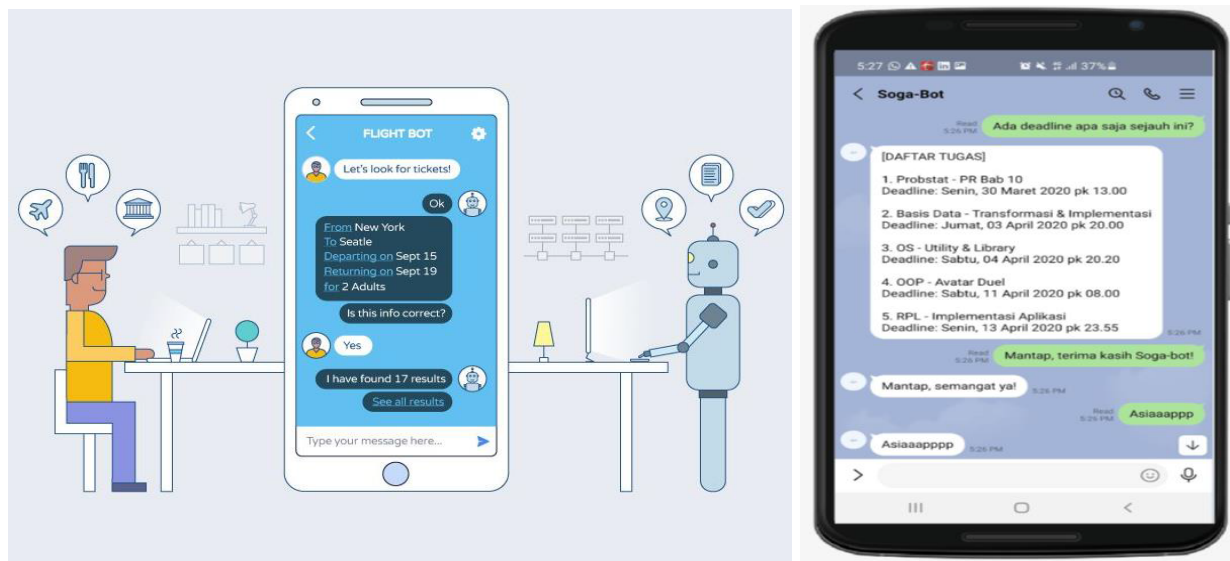
<b>BAB I</b>	
<b>DESKRIPSI TUGAS</b>	<b>2</b>
Latar Belakang	2
Fitur Aplikasi	3
Spesifikasi Program	4
<b>BAB II</b>	
<b>LANDASAN TEORI</b>	<b>5</b>
Algoritma Knuth-Morris-Pratt	5
Algoritma Boyer-Moore	5
Regex	5
Chatbot	5
<b>BAB III</b>	
<b>ANALISIS PEMECAHAN MASALAH</b>	<b>6</b>
Langkah Penyelesaian Setiap Masalah	6
<b>BAB IV</b>	
<b>IMPLEMENTASI &amp; PENGUJIAN</b>	<b>8</b>
<b>BAB V</b>	
<b>KESIMPULAN &amp; SARAN</b>	<b>10</b>
Kesimpulan	10
Saran	10
<b>Daftar Pustaka</b>	<b>11</b>

# 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 Peningkat Deadline.



**Gambar 1. Ilustrasi Chatbot dan Asisten Peningkat 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 Peningkat 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.

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

## 1.2 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** : Suatu kalimat diklasifikasikan sebagai suatu task apabila mengandung tanggal (format dibebaskan), kode mata kuliah / nama mata kuliah (dibebaskan), jenis tugas (berdasarkan daftar kata penting yang sudah disediakan), dan topik tugas (tanpa batasan). Komponen-komponen tersebut diklasifikasikan menggunakan regular expression sehingga masukan kalimat benar-benar layaknya kalimat sehari-hari. 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.
2. **Melihat daftar task yang harus dikerjakan** : Seluruh task yang sudah tercatat oleh assistant, berdasarkan periode waktu tertentu (N minggu ke depan, N hari ke depan, hari ini), berdasarkan jenis task (Sesuai dengan daftar task yang didefinisikan. User dapat melihat daftar task dengan jenis *task* tertentu)
3. **Menampilkan deadline dari suatu task tertentu** : Hanya berlaku untuk task yang bersifat **Tugas** atau memiliki tenggat waktu
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. 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. 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. Bot akan memberikan hasil berupa daftar kata-kata yang bisa digunakan untuk menambahkan dan melihat daftar task (setiap kelompok bebas membentuknya seperti apa)
7. **Mendefinisikan list kata penting terkait apakah itu merupakan suatu task atau tidak** : Minimal terdapat 5 kata penting berbeda, contohnya adalah ["Kuis", "Ujian", "Tucil", "Tubes", "Praktikum"]. Kata penting akan digunakan pada penentuan jenis tugas dari suatu *task*. Daftar kata penting tidak perlu dibuat dinamis, cukup statis saja atau *hardcoded*.

8. **Menampilkan pesan error jika assistant tidak dapat mengenali masukan user :**  
Masukan yang tidak termasuk ke dalam jenis pesan di poin 1 sampai 4 dapat dikategorikan sebagai masukan tak dikenali. Error message dibebaskan sesuai kreativitas mahasiswa
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%. Ada berbagai metrik yang dapat dimanfaatkan untuk mencari kemiripan kata, salah satunya adalah Levenshtein distance yang diukur melalui pendekatan *dynamic programming*.

### 1.3 Spesifikasi Program

1. Aplikasi yang dibuat berbasis web (wajib) dan anda dapat menggunakan salah satu kaskas website: PHP, Flask, Django, JavaScript.
2. Aplikasi (backend) harus menggunakan algoritma pencocokan string KMP, Boyer-Moore, dan Regex dengan menggunakan bahasa yang menunjang regular expression: Java, Javascript, PHP, Python.
3. Penyimpanan data-data dan pengetahuan yang diperlukan oleh Chatbot bisa didefinisikan melalui 2 cara (pilih salah satu), yaitu :
  - a. 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 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

### 2.1 Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt (KMP) mencari pola dalam teks dengan urutan kiri ke kanan (seperti algoritma Brute Force) dengan penggeseran yang lebih cerdas. KMP melakukan pre-proses pada pola tersebut untuk menemukan kecocokan awalan pola dengan pola itu sendiri. *Border function* didefinisikan sebagai ukuran prefiks terbesar dari  $P[0..m]$  yang juga merupakan sufiks dari  $P[1..m]$ . Algoritma KMP memiliki kompleksitas waktu  $O(n + m)$ .

### 2.2 Algoritma Boyer-Moore

Algoritma Boyer-Moore didasarkan pada 2 teknik, yaitu *looking-glass* (mencari P dalam T dengan bergerak mundur melalui P, dimulai dari ujungnya) dan *character-jump* (dilakukan ketika terjadi *mismatch* pada  $T[i] == x$  dan karakter dalam pola  $P[j]$  tidak sama dengan  $T[i]$ ). Jika P berisi x di suatu tempat, coba geser P ke kanan untuk menyelaraskan kemunculan terakhir x di P dengan  $T[i]$ . Jika P berisi x di suatu tempat, tetapi pergeseran ke kanan ke kejadian terakhir tidak dimungkinkan, maka geser P ke kanan sebanyak 1 karakter ke  $T[i + 1]$ . Terakhir, jika 2 kasus sebelumnya tidak terjadi, geser P untuk menyejajarkan  $P[0]$  dengan  $T[i + 1]$ .

### 2.3 Regex

Regex / Regular Expression adalah deretan karakter spesial yang mendefinisikan sebuah pola dalam pencarian teks. Konsep tentang regex pertama kali muncul di tahun 1951, ketika seorang ilmuwan matematika bernama Stephen Cole Kleene memformulasikan definisi tentang *bahasa formal*. Regex dapat digunakan untuk validasi data, pencarian, dan *find & replace*.

### 2.4 Chatbot

Chatterbot (Chatbot) adalah sebuah program komputer berbasis *Artificial Intelligence* (AI) yang dirancang untuk menyimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara audio maupun teks. Teknologi ini juga dikenal sebagai asisten digital yang dapat memahami serta memproses permintaan pengguna, dan memberikan jawaban yang relevan dengan cepat. Itulah mengapa istilah ini dikenal dengan sebutan “bot” yang memang merupakan singkatan dari robot internet. Bahkan, bot itu sendiri dapat menirukan percakapan manusia dalam bentuk teks maupun suara yang diaplikasikan pada website dan aplikasi seperti Facebook, WhatsApp, Twitter dan lain-lain.

## **BAB III**

### **ANALISIS PEMECAHAN MASALAH**

#### **3.1 Langkah Penyelesaian Setiap Masalah**

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

- A. Setiap pesan yang diterima diperiksa menggunakan regex apakah pesan tersebut mengandung kata penting seperti “tugas”, “kuis”, dan lain lain. Apabila pesan tersebut mengandung kata penting, maka dapat dianggap pesan ini adalah pesan menambahkan task
- B. Pesan kemudian kembali diperiksa dengan regex untuk mencari tanggal deadline dari task tersebut.
- C. Jika tanggal ditemukan maka akan dimasukkan kedalam database.
- D. Jika tanggal tidak ditemukan maka dicari kata kunci durasi seperti hari dan angka.
- E. Kata kunci durasi dan angka akan digunakan sebagai penentuan deadline secara relatif

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

- A. Setiap pesan yang diterima diperiksa menggunakan regex apakah pesan tersebut mengandung kombinasi kata kunci “apa” dan “deadline” atau kata kunci apa “apa” dan satu buah kata penting.
- B. Pesan akan kembali diperiksa dengan regex dan dikelompokkan menjadi tipe pesan sesuai dengan spesifikasi
- C. Dilakukan query ke database sesuai dengan kelompok dan kata penting lainnya yang terdapat pada pesan.
- D. Hasil Query dikirim sebagai output chatbot

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

- A. Suatu pesan dikenali sebagai perintah untuk menampilkan deadline dari suatu task tertentu jika terdapat keyword “deadline” dan jenis kata tugasnya
- B. Keyword tersebut akan dicari dengan KMP
- C. Jika ditemukan kecocokan pada database, bot akan menampilkannya

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

- A. Pesan akan dicari kata kunci “deadline” dan pengecekan apakah kata yang ada

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

- A. Pesan akan dicari kunci “selesai” dan ID task atau nama task
- B. Jika kunci ditemukan, task akan dihapus dari database

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

- A. Menampilkan daftar command yang dapat digunakan oleh user

##### **7. Mendefinisikan list kata penting terkait apakah itu merupakan suatu task / tidak**

- A. Kata kunci tersedia hapus, deadline, selesai, tambah, kapan, apa, fitur, bantuan.

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

- A. Pesan akan dicari kata-kata penting yang didefinisikan pada semua fitur lain
- B. Jika tidak ditemukan satupun fitur yang sesuai dengan kata kunci, akan ditampilkan pesan error

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

- A. Kesalahan kata dapat dideteksi dengan mengecek setiap kata pada query/perintah user.
- B. Setiap kata pada perintah dicocokkan dengan kata-kata penting yang telah didefinisikan sebelumnya.
- C. Persentase kemiripan kata pada perintah dengan kata penting diukur dengan :

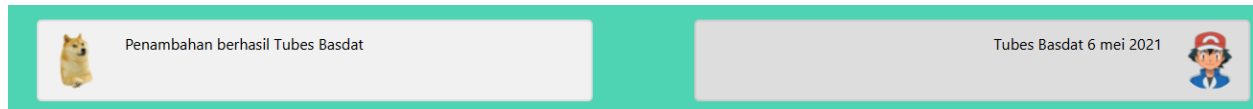
$$\frac{(|a| + |b|) - lev_{a,b}(i, j)}{|a| + |b|}$$

- D. Jika persentase dari kedua kata tersebut lebih dari 75 dan kurang dari 100, akan dikembalikan pesan ke user dengan perbaikan kata yang salah.



## BAB IV IMPLEMENTASI & PENGUJIAN

### 1. Menambahkan task baru



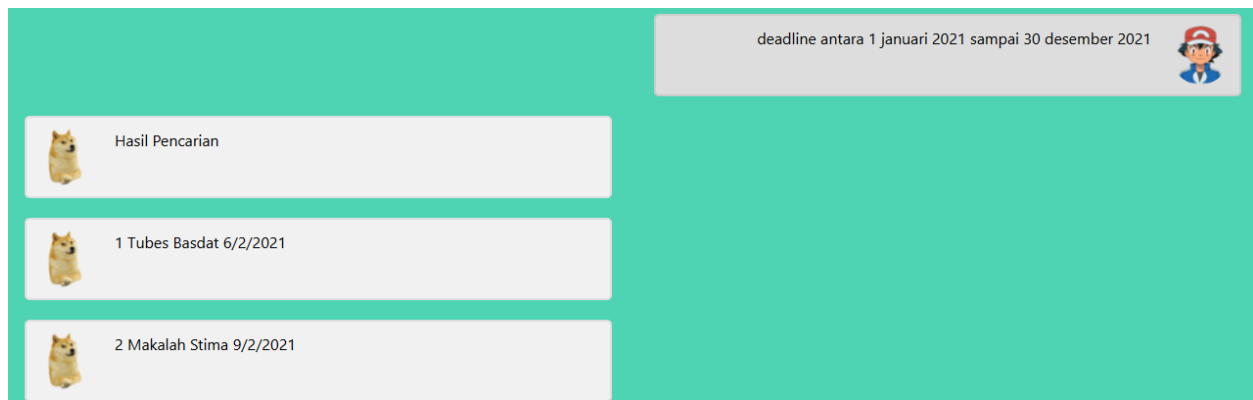
**Gambar 2. Menambahkan task baru**

### 2. Melihat daftar task yang harus dikerjakan



**Gambar 3. Daftar Task**

### 3. Menampilkan deadline dari suatu task tertentu



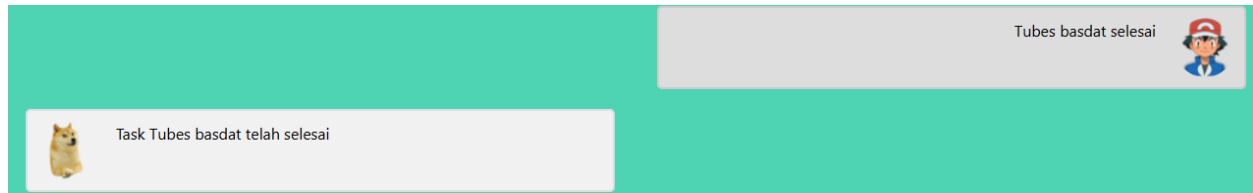
**Gambar 4. Menampilkan Deadline**

### 4. Memperbaharui task tertentu



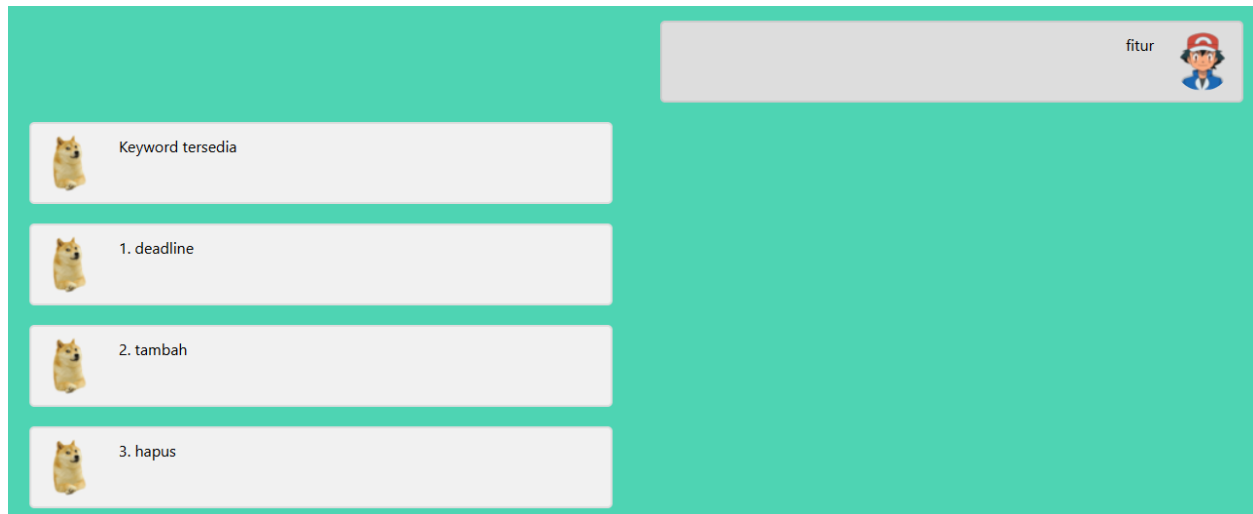
**Gambar 5. Memperbaharui Task**

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



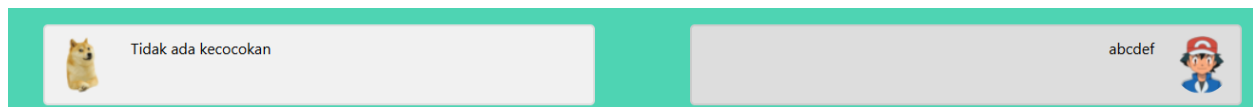
**Gambar 6. Menandai Task yang Selesai**

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



**Gambar 7. Daftar Fitur**

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



**Gambar 8. Pesan Error**

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



**Gambar 9. Dokumentasi**

## **BAB V**

### **KESIMPULAN & SARAN**

#### **5.3. Kesimpulan**

*String Matching* dapat digunakan untuk mengenali pesan dalam bahasa manusia dan menerjemahkannya menjadi bentuk yang diinginkan. Hal tersebut dapat dilakukan dengan memanfaatkan algoritma-algoritma *String Matching* seperti Knuth-Morris-Pratt (KMP) dan Boyer-Moore, atau menggunakan regex. Algoritma KMP dan Boyer-Moore hanya dapat menentukan apakah suatu pola ada di dalam teks atau tidak. Untuk mengakomodasi kemungkinan typo, implementasi *Levenshtein Distance* dapat menjadi solusi untuk menentukan kemungkinan kata yang dimaksud.

#### **5.2. Saran**

- Pengembangan dapat dilakukan dengan memuat lebih banyak kata kunci
- Kepada saudara Tanur mohon segera mengajarkan segala hal yang dia ketahui kepada anggota kelompok ini

## Daftar Pustaka

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>

<https://www.niagahoster.co.id/blog/chatbot-adalah>

<https://www.petanikode.com/regex/>

[https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression)