REVISI

Spesifikasi Tugas Besar III IF2211 Strategi Algoritma
Semester II Tahun 2020/2021

Oleh: M. Rayhan Farrukh - 13523035

Ditulis untuk salah satu tugas seleksi asisten Laboratorium Sistem Terdistribusi STEI ITB
tahun 2025

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

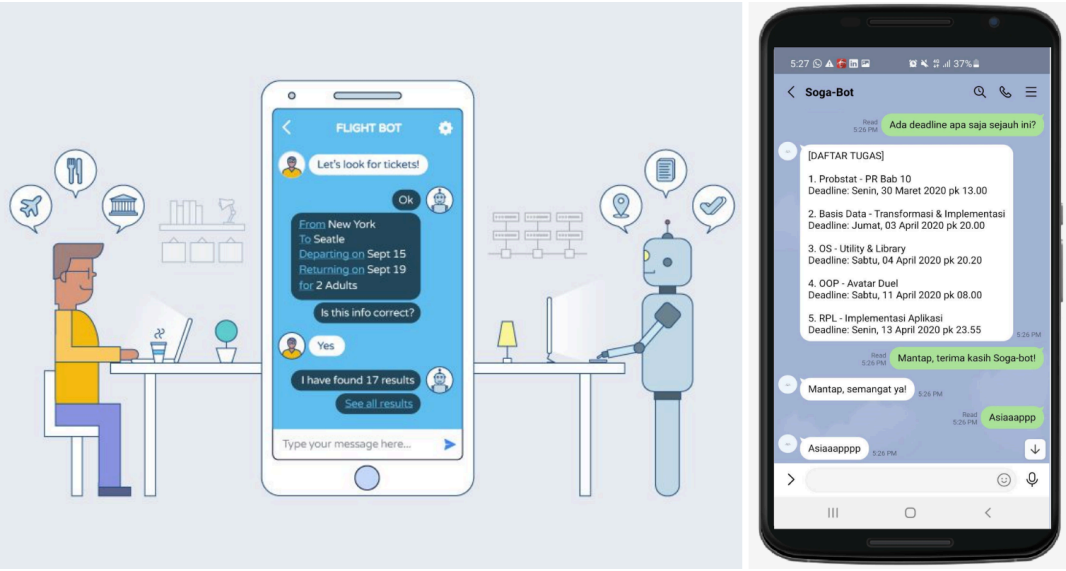
Tugas Besar III IF2211 Strategi Algoritma Semester II Tahun 2020/2021

Penerapan String Matching dan Regular Expression dalam Pembangunan Deadline Reminder Assistant

Batas pengumpulan : Rabu, 28 April 2021 sebelum pukul 23.59 WIB
Arsip pengumpulan :
● *Source* program yang bisa dijalankan disertai readme.txt
● Laporan (soft copy)

Latar belakang

Bukan suatu hal yang janggal jika semakin hari tugas di Teknik Informatika Semester 4 semakin bertambah banyak. Hal ini tentunya mengakibatkan bertambahnya pekerjaan yang harus dilakukan mahasiswa. Tak jarang pula ada tugas yang terlupakan karena mahasiswa kesulitan untuk mengingat semua tugas beserta *deadline*-nya. Oleh karena itu, mahasiswa Teknik Informatika berniat untuk membuat suatu Google Assistant sederhana berupa Deadline Reminder Assistant, atau dalam bahasa Indonesia 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 *chatbot* seperti 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 menyajikan informasi yang diperlukan dan menjawab berbagai pertanyaan yang 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, atau task-task penting lainnya yang perlu dilakukan. *Chatbot* ini akan sangat membantu *user* agar tidak lagi melewatkan *deadline* tugas.

Deskripsi tugas

Dalam tugas besar ini, Anda akan diminta untuk membangun sebuah *chatbot* sederhana yang berfungsi untuk membantu *user* mengingat berbagai *deadline*, tanggal penting, serta *task-task* tertentu. Dengan memanfaatkan algoritma *string matching* dan *regular expression*, Anda dapat membangun sebuah *chatbot* interaktif sederhana layaknya Google Assistant yang dapat menjawab segala pertanyaan Anda terkait informasi *deadline* tugas-tugas yang ada.

Fitur-Fitur Aplikasi

Deadline Reminder Assistant akan dibangun dengan sistem ***question and answer*** dimana pengembang diharapkan sudah menyediakan kumpulan formula tertentu untuk mendeteksi setiap perbedaan *command* pada aplikasi *chatbot*. Berikut runtutan fitur yang harus dimiliki oleh Deadline Reminder Assistant tersebut.

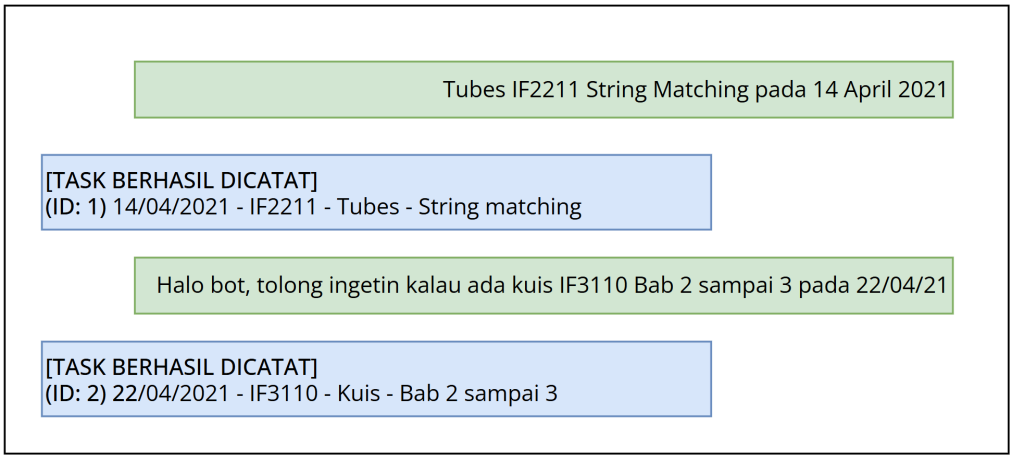
1. Menambahkan *task* baru
   Untuk menambahkan tugas baru, kalimat perintah dari pengguna harus memenuhi kriteria berikut:
   a. Suatu kalimat diklasifikasikan sebagai sebuah *task* apabila mengandung **semua komponen** berikut ini:
      - Tanggal (format tanggal dibebaskan)
      - Kode Mata Kuliah / Nama Mata Kuliah (pilih salah satu)
      - Jenis Tugas (berdasarkan daftar kata penting yang sudah disediakan)
      - Topik Tugas (tidak ada batasan)

   b. Point i sampai iv diklasifikasikan menggunakan ***regular expression*** sehingga masukan kalimat benar-benar menyerupai 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

Contoh Interaksi



Gambar 2. Interaksi fitur menambahkan *task*
Sumber: Dokumentasi Pribadi

2. Melihat daftar task yang harus dikerjakan
   Daftar *task* yang ditampilkan bisa dikelompokkan berdasarkan beberapa kriteria berikut:

   a. Seluruh task yang sudah tercatat oleh *assistant*
      - Contoh perintah: "Apa saja *deadline* yang dimiliki sejauh ini?"

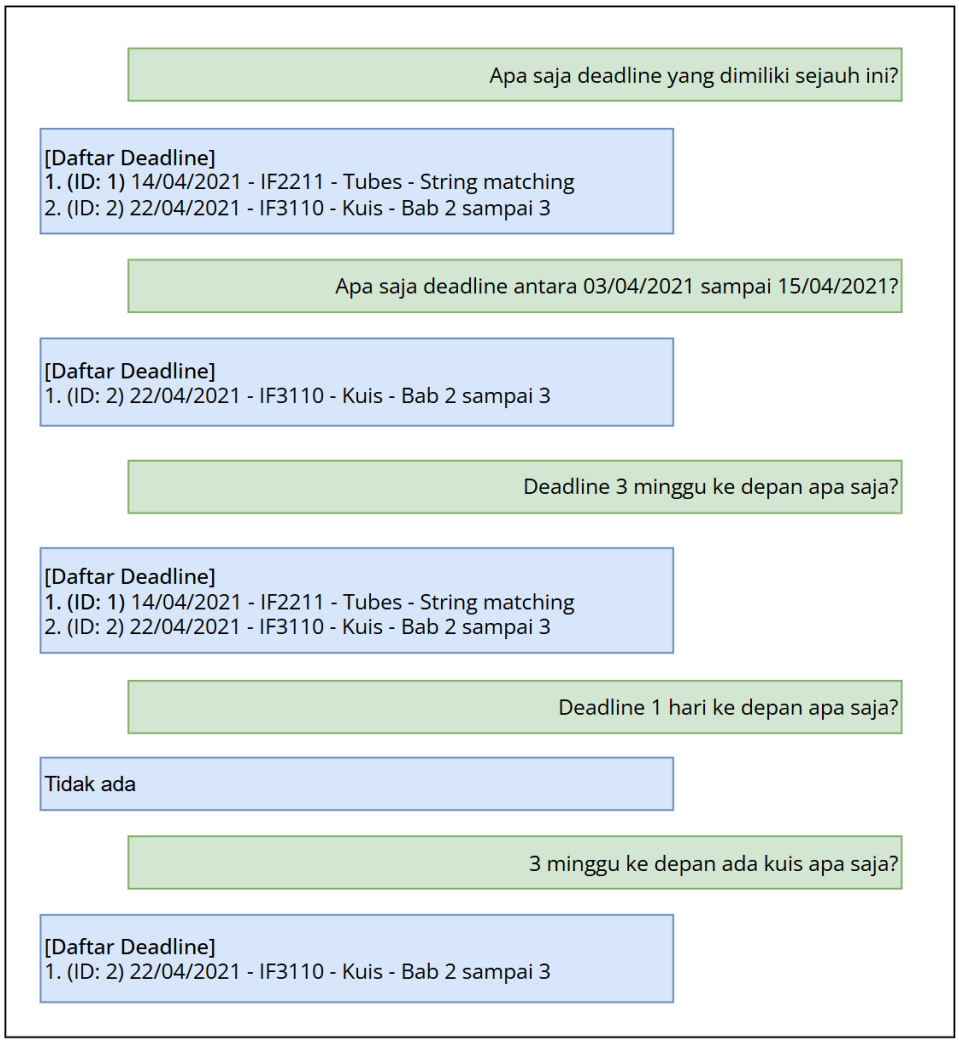
   b. Berdasarkan periode waktu
      - Pada periode tertentu (DATE_1 sampai DATE_2)
        Contoh perintah: "Apa saja *deadline* antara DATE_1 sampai DATE_2?"
      - N minggu ke depan
        Contoh perintah: "*Deadline* N minggu ke depan apa saja?"
      - N hari ke depan
        Contoh perintah: "*Deadline* N hari ke depan apa saja?"
      - Hari ini
        Contoh perintah: "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 tiga minggu kedepan

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

Contoh Interaksi



Gambar 3. Interaksi fitur melihat daftar *task*
Sumber: Dokumentasi Pribadi

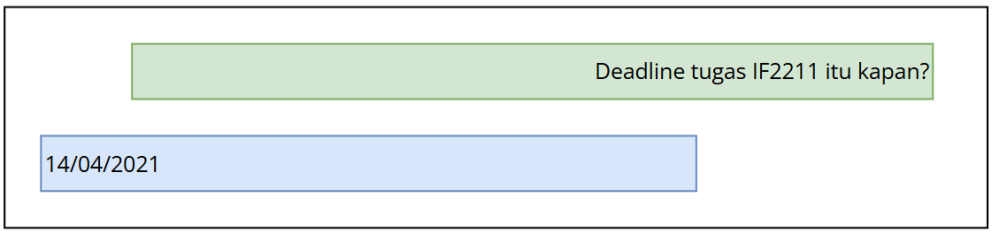
Keterangan penting:

- Perintah yang digunakan pengguna tidak harus sama, asalkan mengandung kata kunci yang ditentukan (kata kunci tiap perintah bisa ditentukan sendiri). Misalnya, kedua contoh di bawah ini memberikan output yang sama
    - Apa saja ***deadline*** antara **03/04/2021** sampai **15/04/2021**?

- Antara **<u>03/04/2021</u>** dan **<u>15/04/2021</u>** ada ***<u>deadline</u>*** 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



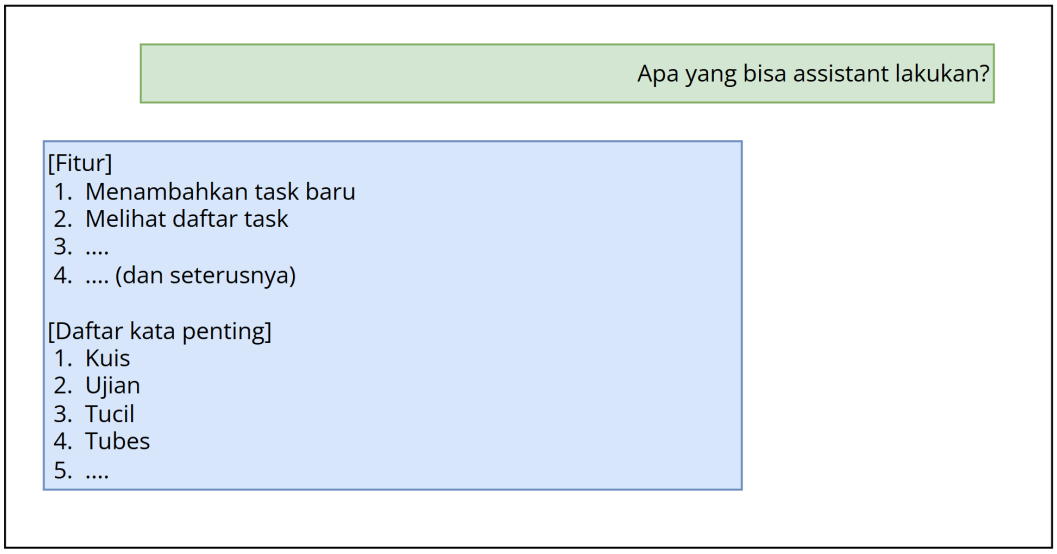
Gambar 4. Interaksi fitur menampilkan *deadline* tugas
Sumber: Dokumentasi Pribadi

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 satu *keyword* untuk memperbarui suatu task dan nomor task tertentu. Misalnya:
      - "*Deadline task* X diundur menjadi 28/04/2021" dimana X merupakan nomor ID dari suatu *task*.
   c. 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 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*.
   b. 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

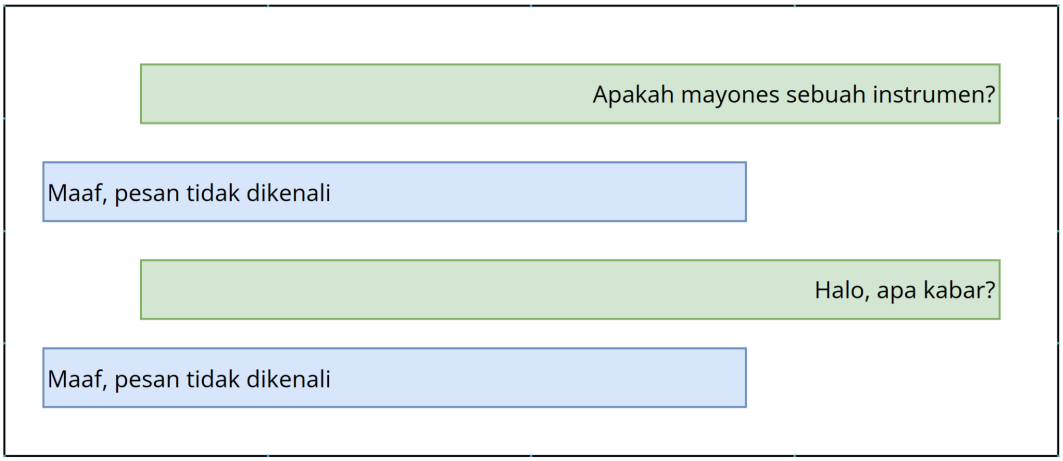


Gambar 5. Interaksi fitur menampilkan *deadline* tugas
Sumber: Dokumentasi Pribadi

7. Mendefinisikan sebuah *list* sebagai daftar kata kunci untuk mengidentifikasi jenis tugas, dengan ketentuan berikut:
   a. Minimal terdapat lima kata penting berbeda, contohnya:
      [“Kuis”, “Ujian”, “Tucil”, “Tubes”, “Praktikum”]
   b. Daftar kata penting tidak perlu dibuat dinamis, cukup *static* saja atau *hardcoded*.

8. Menampilkan pesan error jika assistant tidak dapat mengenali masukan user. Masukan yang tidak termasuk ke dalam jenis pesan pada poin 1 sampai 4 dapat dikategorikan sebagai masukan yang tak dikenal, dan *assistant* harus memberi tahu *user* tentang itu. *Error message* dibebaskan, sesuai kreativitas mahasiswa

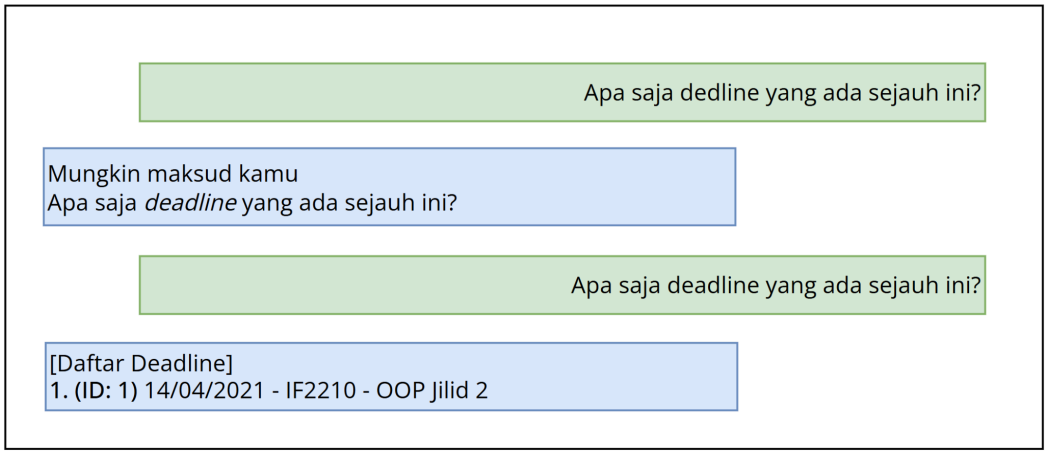
Contoh interaksi



Gambar 6. Interaksi fitur pesan *error*
Sumber: Dokumentasi Pribadi

9. **(Bonus)** *Chatbot* dapat memberikan rekomendasi kata jika terdapat kesalahan ejaan (*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



Gambar 7. Interaksi pendeteksian *typo*
Sumber: Dokumentasi Pribadi

Ada berbagai metriks yang dapat dimanfaatkan untuk mencari kemiripan kata, salah satunya adalah Levenshtein *distance* yang diukur melalui pendekatan *dynamic programming*. Anda dapat mempelajari Levenshtein *distance* melalui pranala ini.

Spesifikasi Program

1. Aplikasi yang dibuat berbasis web (wajib) dan Anda dapat menggunakan salah satu kakas *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 dan pengetahuan yang diperlukan oleh *chatbot* bisa didefinisikan melalui dua 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 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 mencocokkan kata kunci dengan input *user*.
   b. Rekomendasi kata: pencocokan *fuzzy matching* (Levenshtein *distance*) dapat dimanfaatkan untuk menentukan tingkat kemiripan suatu kata pada perintah.
   c. Pengekstrakan nilai-nilai berjenis numerik dan tanggal dilakukan dengan memanfaatkan *regular expression*.

Lain-lain

1. Anda dapat menambahkan fitur-fitur lain yang menunjang program yang dibuat (unsur kreativitas).

2. Tugas dikerjakan berkelompok, minimal 2 orang dan maksimal 3 orang, boleh lintas kelas namun tidak boleh sekelompok dengan orang yang sama pada tubes stima sebelumnya.
3. Semua kelompok harap mengisi data kelompok pada *link* http://tiny.cc/PendataanStima3.
4. Anda harus membuat aplikasi dan program ini sendiri kecuali *library file* dan *regex*, tetapi belajar dari contoh-contoh program serupa yang sudah ada tidak dilarang. Tidak boleh melakukan plagiasi *source code* dari program orang lain, meskipun itu teman Anda dari kelompok lain.
5. Program harus modular dan mengandung komentar yang jelas.
6. Beri nama tokoh untuk *chatbot* ini, misalnya Sogabot, Hansbot, Awoobot, dll. Lengkapi dengan gambar avatarnya. Avatar yang dinamis (bibir bisa bergerak, gerakan tubuh) adalah kreativitas yang dihargai.
7. Batas akhir pengumpulan tugas adalah Rabu, 28 April 2021 pukul 23.59 WIB. Keterlambatan dalam pengumpulan akan diberi penalti pengurangan skor yang cukup signifikan.
8. Semua pertanyaan menyangkut tugas ini dapat dikomunikasikan lewat QnA yang bisa diakses pada *link* http://tiny.cc/QnATubes3
9. **Bonus (maksimal 15 poin)**:
    a. Men-*deploy* aplikasi web yang telah dibangun (*hosting provider* dibebaskan). *Deployment website* harus dipertahankan sampai demo tugas besar.
    b. Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke Youtube. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi, mahasiswa akan diminta memutar video Youtube tersebut di depan asisten. Beberapa contoh video tubes tahun-tahun sebelumnya dapat dilihat di YouTube dengan menggunakan kata kunci “Tubes Stima”, “Tugas besar stima”, “strategi algoritma”, dll.
    c. Membuat fitur rekomendasi kata untuk meng-*handle* ketika terdapat kesalahan kata pada perintah.
10. Akan diadakan demo, tunggu informasi lebih lanjut setelah waktu pengerjaan tugas berakhir.
11. Setiap anggota kelompok harus memahami seluruh program, termasuk bagian yang tidak mereka kerjakan.

12. Program disimpan dalam folder "**Tubes3_NIM**" dengan NIM merupakan NIM anggota terkecil. Berikut struktur dari isi folder tersebut:
    a. Folder "**src**" berisi ***source code***
    b. Folder "**doc**" berisi **laporan tugas besar** dengan format "**nama_kelompok.pdf**"
    c. Folder "**test**" berisi **data** atau **pengetahuan** awal yang dimiliki oleh *chatbot*. Apabila menggunakan *database* lokal, hasil *dump*-nya dapat disimpan disini.
    d. README selengkap mungkin. Referensi README dapat diakses pada https://github.com/ritaly/README-cheatsheet atau referensi lain yang serupa.
13. Folder tersebut **di-*zip*** dengan format yang sama dengan nama folder. Link pengumpulan akan diberitahukan lebih lanjut oleh asisten.

Isi laporan

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

Keterangan laporan

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

Penilaian

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

--- *Selamat Mengerjakan, it's not worth it if you're not having fun* ---