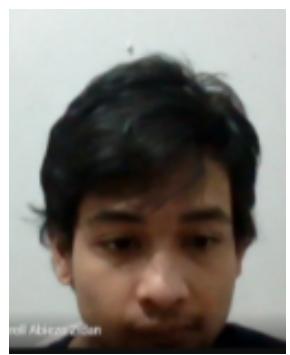


**LAPORAN TUGAS BESAR 3**  
**IF2211 – STRATEGI ALGORITME**  
**SEMESTER II - TAHUN 2020/2021**

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



**1**

**2**

**3**

Daftar Anggota Kelompok:

- 1. Leonardus Brandon Luwianto (13519102)**
- 2. Made Kharisma Jagaddhita (13519176)**
- 3. Farrell Abieza Zidan (13519182)**

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
2021

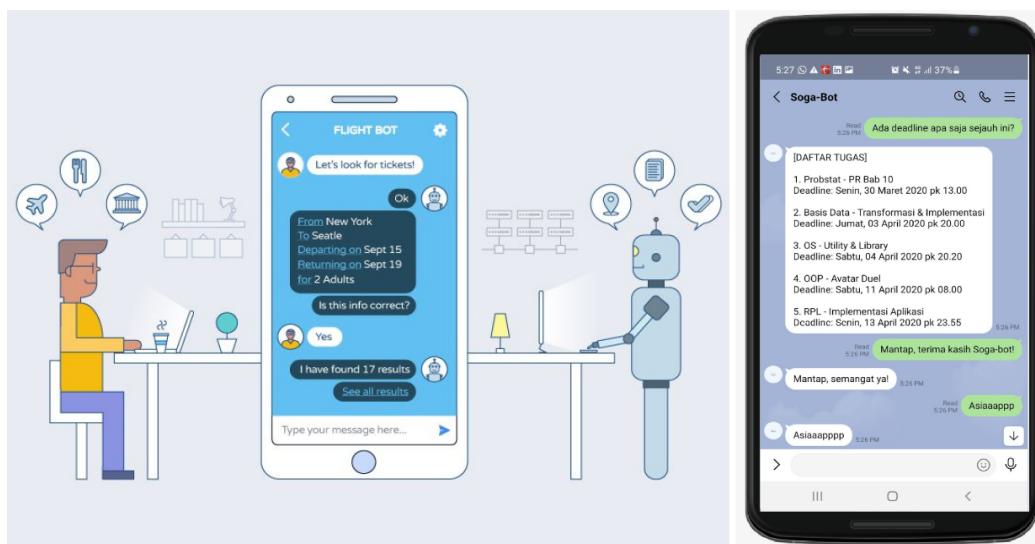
## **DAFTAR ISI**

<b>DAFTAR ISI</b>	<b>2</b>
<b>BAB I</b>	<b>3</b>
<b>BAB II</b>	<b>6</b>
Deskripsi Singkat Algoritma KMP, BM, dan Regex	6
Algoritma Knuth-Morris-Pratt (KMP)	6
Algoritma Boyer-Moore (BM)	7
Regular Expression (Regex)	8
Penjelasan Singkat mengenai Chatbot	9
<b>BAB III</b>	<b>11</b>
Langkah Penyelesaian Masalah Setiap Fitur	11
Fitur Fungsional dan Arsitektur Chatbot yang Dibangun	13
<b>BAB IV</b>	<b>30</b>
Spesifikasi Teknis Program	30
Penjelasan Tata Cara Penggunaan Program	32
Hasil Pengujian	32
Halaman Awal	32
Analisis Hasil Pengujian	38
<b>BAB V</b>	<b>39</b>
Kesimpulan	39
Saran	39
Refleksi dan Komentar terhadap Tugas	39
<b>DAFTAR PUSTAKA</b>	<b>40</b>

# BAB I

## 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.



Gambar 1.1 Ilustrasi chatbot dalam aplikasi berbasis web

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.

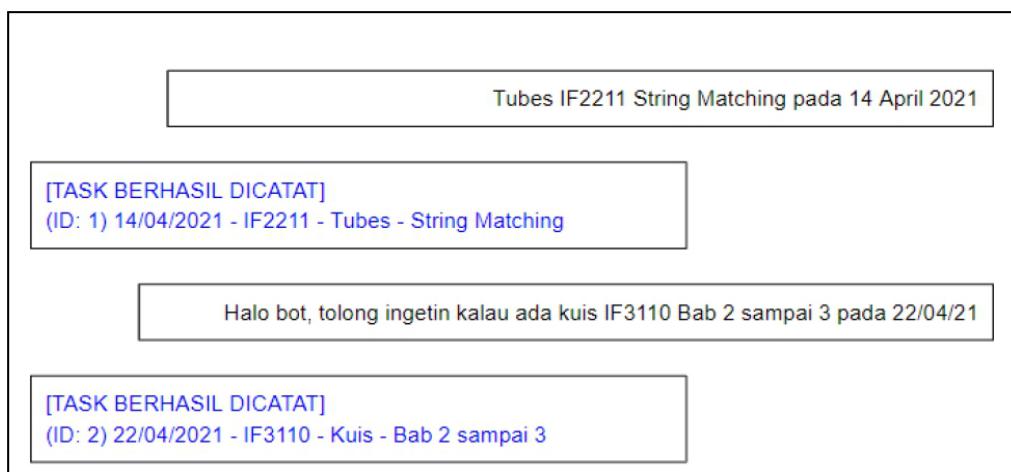
1. Menambahkan task baru
  - a. Suatu kalimat diklasifikasikan sebagai suatu task apabila mengandung semua komponen berikut ini:
    - i. Tanggal (format dibebaskan)
    - ii. Kode Mata Kuliah / Nama Mata Kuliah (dibebaskan)
    - iii. Jenis Tugas (berdasarkan daftar kata penting yang sudah disediakan)

- iv. 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



Gambar 1.2 Contoh interaksi antara inputan user dan keluaran dari chatbot

## 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
  - i. Pada periode tertentu (DATE\_1 until DATE\_2)
    - Contoh perintah yang dapat digunakan: “Apa saja deadline antara DATE\_1 sampai DATE\_2?”

ii. N minggu ke depan

-Contoh perintah yang dapat digunakan: “Deadline N minggu ke depan apa saja?”

iii. N hari ke depan

-Contoh perintah yang dapat digunakan: “Deadline N hari ke depan apa saja?”

iv. Hari ini

-Contoh perintah yang dapat digunakan: “Apa saja deadline hari ini?”

c. Berdasarkan jenis task (kata penting)

i. Sesuai dengan daftar task yang didefinisikan

ii. User dapat melihat daftar task dengan jenis task tertentu

iii. Misalnya: “3 minggu ke depan ada kuis apa saja?”, maka Chatbot akan menampilkan daftar kuis selama 3 minggu kedepan

## BAB II

# LANDASAN TEORI

### 2.1 Deskripsi Singkat Algoritma KMP, BM, dan Regex

Algoritma Knuth-Morris-Pratt (KMP), Boyer-Moore (BM), dan *Regular Expression* (Regex) merupakan algoritma string pattern atau pattern matching yang digunakan dalam mencari string tertentu dalam suatu text. Pattern matching itu sendiri sebenarnya dapat diimplementasikan dengan menggunakan algoritma brute force, namun dalam hampir seluruh kasus, algoritma brute force jarang digunakan dalam pattern matching karena kurang efisien dalam pencarinya, terutama dalam pencarian text yang bisa jadi kalimatnya mencapai lebih dari satu halaman.

KMP, BM, dan Regex merupakan improvisasi dari algoritma brute force dimana pencarian sebuah key tidak dilakukan secara satu per satu namun dengan pendekatan yang lebih intelejen dan juga masih menjamin untuk mendapatkan solusi (jika memang keyword tersebut ada pada text). Algoritma tersebut mengurangi beberapa langkah pencarian keyword yang secara logika sebenarnya tidak perlu dicari karena langkah tersebut tidak akan pernah menuju kepada solusi.

#### 2.1.1 Algoritma Knuth-Morris-Pratt (KMP)

Knuth-Morris-Pratt merupakan algoritma pintar yang memiliki metode lebih complex dibanding brute force. Algoritma KMP mengecek sebuah string dari kiri ke kanan namun tidak setiap baris dilakukan pengecekan satu per-satu.

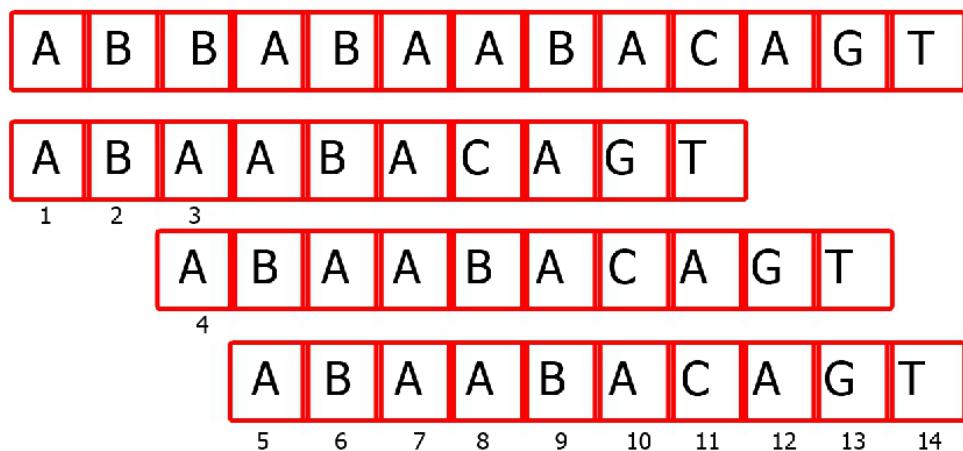
Algoritma KMP memiliki konsep untuk menggeser string sejauh n ketika ditemukan perbedaan karakter, dengan n adalah nilai dari prefix dan sufix terpanjang yang sama pada saat pengecekan berada di posisi k string input. Itu berarti, pada algoritma KMP dibuat border function sebelumnya, yaitu berupa table prefix suffix.

Misalkan input berupa string “ABAABACAGT” maka untuk setiap ukuran string, perlu dicari prefix dan sufix terpanjangnya, yaitu :

- “A” memiliki prefix & sufix terpanjang 0,
- “AB” memiliki prefix & sufix terpanjang 0,
- “ABA” memiliki prefix & sufix terpanjang 1 (yaitu “A”),

“ABAA” memiliki prefix & suffix terpanjang 1 (yaitu “A”),  
 “ABAAB” memiliki prefix & suffix terpanjang 2 (yaitu “AB”),  
 “ABAABA” memiliki prefix & suffix terpanjang 3 (yaitu “ABA”),  
 “ABAABAC” memiliki prefix & suffix terpanjang 0,  
 “ABAABACA” memiliki prefix & suffix terpanjang 0,  
 “ABAABACAG” memiliki prefix & suffix terpanjang 0,  
 “ABAABACAGT” memiliki prefix & suffix terpanjang 0,

Karena semua prefix dan suffix sudah diketahui, sekarang pencarian pattern bisa dilakukan, anggap string yang dicari adalah “ABBABAABACAGT”, maka Langkah untuk pencarian dengan algoritma KMP adalah sebagai berikut :



Gambar 2.1 Langkah penyelesaian string pattern matching dengan KMP

1. Cek apakah text index 1 = A, hasilnya benar, lanjut pengecekan berikutnya
2. Cek apakah text index 2 = B, hasilnya benar, lanjut pengecekan berikutnya
3. Cek apakah text index 3 = B, hasilnya salah, geser sebanyak 2 kali (ingat prefix dan suffix)
4. Cek apakah text index 3 = B, hasilnya salah, geser sebanyak 1 kali
5. Cek apakah text index 4 = A, hasilnya benar, lanjut pengecekan berikutnya
6. Cek apakah text index 5 = B, hasilnya benar, lanjut pengecekan berikutnya
7. Cek apakah text index 6 = A, hasilnya benar, lanjut pengecekan berikutnya
8. Cek apakah text index 7 = A, hasilnya benar, lanjut pengecekan berikutnya
9. Cek apakah text index 8 = B, hasilnya benar, lanjut pengecekan berikutnya
10. Cek apakah text index 9 = A, hasilnya benar, lanjut pengecekan berikutnya
11. Cek apakah text index 10 = C, hasilnya benar, lanjut pengecekan berikutnya
12. Cek apakah text index 11 = A, hasilnya benar, lanjut pengecekan berikutnya
13. Cek apakah text index 12 = G, hasilnya benar, lanjut pengecekan berikutnya
14. Cek apakah text index 13 = T, hasilnya benar, string ditemukan

### 2.1.2 Algoritma Boyer-Moore (BM)

Boyer-Moore merupakan algoritma yang efisien sama seperti KMP dalam pencarian string pattern, namun dilihat dari performa, jika KMP memiliki best case ketika pattern yang ada memiliki digit yang keberagamannya sedikit, sedangkan algoritma BM memiliki best case ketika pattern memiliki simbol yang beragam.

BM memiliki 2 konsep dasar, yaitu pencarian dilakukan dari sebelah kanan (terbalik), dan teknik pergeseran pattern nya. Dalam kasusnya sendiri terbagi menjadi 3, yaitu :

1. Jika dalam pengecekan string, karakter adalah X dan tidak cocok dengan pattern, dan terdapat karakter X dalam sebelah kiri pattern saat ini, maka geser sampai X dalam pattern sejajar dengan X dalam string.

- “A B A C X A B S”
- “D A X A C“

Dalam kasus ini ketika kita mengecek apakah C = X, karena tidak sesuai maka kita geser seperti ini :

- “A B A C X A B S”
- \_\_\_\_\_ “D A X A C“

2. Jika dalam pengecekan string, karakter adalah X dan tidak cocok dengan pattern, dan terdapat karakter X dalam sebelah kanan pattern saat ini, maka geser pattern sejauh 1

- “A B C A C A B S”
- “D A P A C“

Dalam kasus ini ketika kita mengecek apakah C = P, karena tidak sesuai maka kita geser seperti ini :

- “A B C A C A B S”
- \_\_\_\_\_ “D A P A C“

3. Jika kasus pertama dan kedua tidak terpenuhi dalam pengecekan string, maka pergeseran pattern dilakukan sampai index pada string tersebut dilalui

- “A B G A C A B S Q R S T U”
- “D A Z A C“

Dalam kasus ini ketika kita mengecek apakah G = Z, karena tidak sesuai maka kita geser seperti ini :

- “A B G A C A B S Q R S T U”
- \_\_\_\_\_ “D A Z A C“

### 2.1.3 *Regular Expression (Regex)*

Regex akan lebih berguna jika suatu pencarian pattern dilakukan tidak secara eksak. Maksud eksak disini adalah keyword yang benar benar sama persis dengan string yang dicocokkan, dilihat dari kapital atau tidaknya huruf, terdapat spasi atau tidak dan lain sebagainya. Dengan menggunakan regex, pencarian bisa saja menemukan hasil pattern yang tidak begitu sama persis namun masih mendapat keyword penting yang menjadi dasar pencarian tersebut.

Regex menjadi dasar dari bagaimana chatbot bekerja. Dibanding dengan algoritma KMP dan BM, chatbot cenderung diterapkan dengan menggunakan regex karena inputan tidak perlu begitu eksak dan less-strict sehingga bot lebih memahami banyak bentuk keyword yang sebetulnya memang memiliki arti sama. Meskipun demikian, algoritma KMP dan BM pun tetap layak pula diterapkan dalam pembuatan chatbot.

Regex memiliki simbol simbol tertentu yang memberikan instruksi kepada program apa sajakah karakter yang diperbolehkan / dianggap cocok dalam pencocokan

string saat itu. Simbol tersebut menjadi dasar mengapa regex disebut sebagai approximation string matching dan bukan exact string matching.

The screenshot shows a table of regular expression symbols with their meanings. The table has two columns: the symbol on the left and its description on the right. Some rows have a yellow background, while others have a white background. A blue 'Print' icon and a red 'Close' icon are at the top right. At the bottom right, there is a link 'Near-complete reference'.

.	Any character except newline.
\.	A period (and so on for \*, \(), \\", etc.)
^	The start of the string.
\$	The end of the string.
\d,\w,\s	A digit, word character [A-Za-z0-9_], or whitespace.
\D,\W,\S	Anything except a digit, word character, or whitespace.
[abc]	Character a, b, or c.
[a-z]	a through z.
[^abc]	Any character except a, b, or c.
aa bb	Either aa or bb.
?	Zero or one of the preceding element.
*	Zero or more of the preceding element.
+	One or more of the preceding element.
{n}	Exactly n of the preceding element.
{n,}	n or more of the preceding element.
{m,n}	Between m and n of the preceding element.
??,*?,+?,{n}?, etc.	Same as above, but as few as possible.
(expr)	Capture expr for use with \1, etc.
(?:expr)	Non-capturing group.
(?=expr)	Followed by expr.
(?!expr)	Not followed by expr.

Gambar 2.2 Simbol / Notasi umum pada Regular Expression

Sumber :

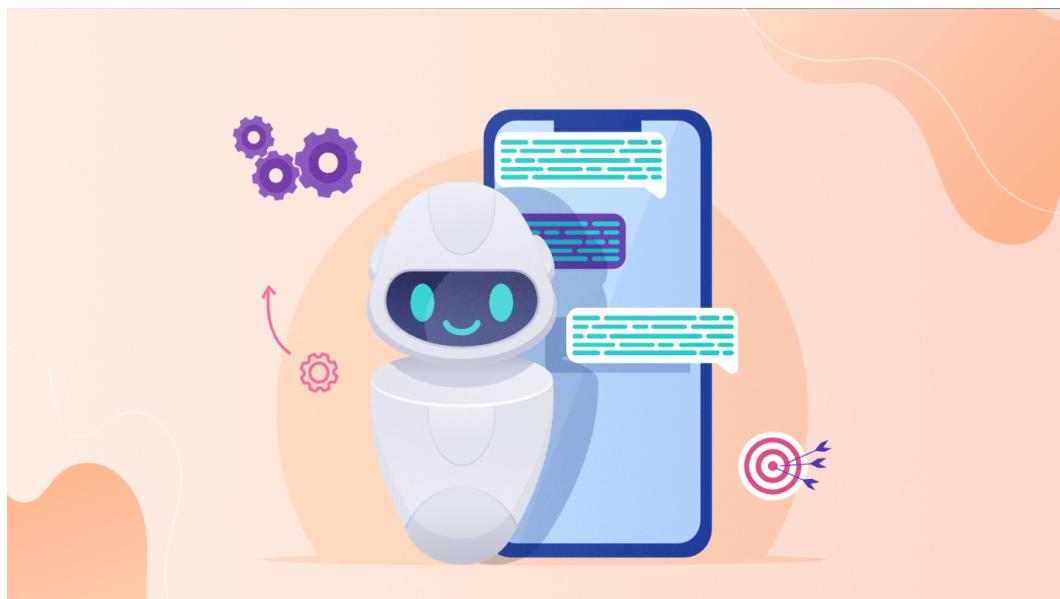
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regular-Expression-2019.pdf>

## 2.2 Penjelasan Singkat mengenai *Chatbot*

Chatbot merupakan bot / AI yang mampu berinteraksi dengan user melalui chat. Chatbot biasanya mengaplikasikan konsep machine learning dimana bot dapat mempelajari sebuah kalimat kalimat baru dan menjawabnya secara tepat dan akurat, dan kemudian menyimpan semua record chat kedalam database karena dalam suatu waktu kalimat tersebut bisa saja ditanya oleh user kembali. Namun chatbot yang hanya mengaplikasikan database tanpa machine learning juga umum dijumpai saat ini.

Chatbot memiliki banyak aplikasi untuk beragam aplikasi digital saat ini. Website bisnis dan app yang berhubungan langsung dengan customer sebenarnya kerap menggunakan chatbot yang berguna untuk menginstruksi customer dan melayaninya terlebih dahulu sebelum service dari staff benar benar dilakukan. Namun belakangan ini chatbot juga ditemui dalam bentuk hiburan dan entertainment untuk usernya ketika merasa jemu dan membutuhkan kebutuhan sosialnya terpenuhi.

Dan dalam beberapa hal, chatbot entertainment ini tidak jarang juga dibuat sebagai bot asisten seperti pada tugas kali ini misalnya.



Gambar 2.3 Ilustrasi mengenai chatbot dalam sebuah web based app

Secara algoritma, chatbot menerapkan algoritma perbandingan string atau pattern matching. Algoritma pattern matching dalam chatbot membandingkan sebuah kalimat yang berasal dari input user sebagai chat user yang kemudian string tersebut dibandingkan dalam database chatbot yang sudah dibuat sebelumnya. Perbandingan bisa saja menggunakan algoritma Knuth-Morris-Pratt(KMP), Atau Boyer-Moore(BM), atau regex. Setelah ditemukan string yang relevan, bot akan menganggap string tersebut menanyakan sesuatu yang dapat dijawab dengan sesuatu yang sudah berada di dalam database bot, sebagai contoh jika string user berisi “Sekarang jam berapa?”, bot akan mencari keyword dalam database untuk menjawab kalimat tersebut. Anggap “jam” merupakan key yang sudah dikenal dalam database, maka seharusnya bot paham dengan pertanyaan tersebut dan menjawabnya dengan sesuatu yang berhubungan dengan waktu.

## BAB III

### ANALISIS PEMECAHAN MASALAH

#### 3.1 Langkah Penyelesaian Masalah Setiap Fitur

Semua hal yang berkaitan dengan menampilkan dan mengubah task akan dilakukan memodifikasi relasi tasks pada database.

##### 1. Menambahkan task baru

- Ambil semua kata penting untuk penambahan task baru. Pada database terdapat pada kolom keyword dari relasi new\_task\_keywords.
- Untuk setiap keyword, masukkan keyword ke dalam regular expression. Regular expression yang terbentuk:

```
/$new_task_keywords[$i]\s*[A-Za-z]{2}\d{4}\s*(.*)\s*pada\s*\d{2}\/\d{2}\/\d{4}/i
```

Arti dari regular expression diatas adalah:

\$new\_task\_keywords[\$i] yaitu kata kunci pembentukan task baru indeks ke-i.  
\s\* adalah tanda spasi berapapun.

[A-Za-z]{2}\d{4} adalah kode mata kuliah, yang terdiri dari 2 huruf dan 4 angka tanpa dipisah dengan spasi.

(.\*) Akan mengambil kata apa saja yang berada di antara kode mata kuliah dan kata ‘pada’. Nantinya akan diekstrak menjadi topik dari task.

\d{2}\/\d{2}\/\d{4} Untuk mengambil tanggal dengan format dd/mm/yyyy.

/i Agar menjadikan regex case insensitive.

Sehingga, untuk menambahkan task baru dilakukan dengan menjalankan perintah <any text> <tipe\_tugas> <kode matkul> <topik> pada <tanggal> <any text>

- Jika pada kata yang diketik terdapat kata yang cocok, maka ekstrak data-data tipe task, kode mata kuliah, topik, dan tanggal deadlinenya. Setelah itu masukkan ke dalam database
2. a. Menampilkan seluruh daftar task yang tercatat sejauh ini
- Pertama program akan mencari kata ‘deadline’ atau ‘task’ pada kata yang dimasukkan. Jika ada, maka dilanjutkan dengan mencari kata ‘sejauh ini’ atau ‘saat ini’ pada kata. Jika ada, maka bot akan menampilkan seluruh task yang ada. Seluruh pencarian dilakukan dengan metode string matching KMP
- b. Menampilkan task berdasarkan periode waktu
- Pertama, program akan mencari kata ‘deadline’ atau ‘task’. Pencarian dilakukan dengan metode string matching KMP
  - Jika ada, maka dilanjutkan dengan mencari 2 format tanggal pada kata yang dimasukkan. Pencarian dilakukan dengan regex. Jika ada 2 format tanggal, maka akan

menampilkan semua task yang ada di antara tanggal yang muncul terlebih dahulu dan muncul setelahnya. Jika tidak ada, maka lanjut ke langkah selanjutnya.

- Program akan mencari kata ‘N minggu ke depan’ pada kata yang dimasukkan dengan N adalah integer. Pencarian dilakukan dengan regex. Jika ada kata yang cocok, maka akan menampilkan semua task N minggu ke depan. Jika tidak ada, maka lanjut ke langkah selanjutnya.
  - Program akan mencari kata ‘N hari ke depan’ pada kata yang dimasukkan dengan N adalah integer. Pencarian dilakukan dengan regex. Jika ada kata yang cocok, maka akan menampilkan semua task N hari ke depan. Jika tidak ada, maka lanjut ke langkah selanjutnya.
  - Program akan mencari kata ‘hari ini’ pada kata yang dimasukkan. Pencarian dilakukan dengan string matching KMP. Jika ada kata yang cocok, maka akan menampilkan semua task pada hari ini.
- c. Menampilkan task berdasarkan berdasarkan jenis task
- Program akan mengambil semua kata penting untuk jenis task. Pada database terdapat pada kolom keyword dari relasi new\_task\_keywords.
  - Program akan mencari 2 format tanggal pada kata yang dimasukkan. Pencarian dilakukan dengan regex. Jika ada 2 format tanggal, maka akan mencari jenis task pada kata. Pencarian dilakukan dengan string matching KMP. Jika ada, maka akan menampilkan semua task dengan jenis task yang sesuai yang ada di antara tanggal yang muncul terlebih dahulu dan muncul setelahnya. Jika tidak ada, maka lanjut ke langkah selanjutnya.
  - Program akan mencari kata ‘N minggu ke depan’ pada kata yang dimasukkan dengan N adalah integer. Pencarian dilakukan dengan regex. Jika ada kata yang cocok, maka akan mencari jenis task pada kata. Pencarian dilakukan dengan string matching KMP. Jika ada, akan menampilkan semua task N minggu ke depan. Jika tidak ada, maka lanjut ke langkah selanjutnya.
  - Program akan mencari kata ‘N hari ke depan’ pada kata yang dimasukkan dengan N adalah integer. Pencarian dilakukan dengan regex. Jika ada kata yang cocok, maka akan mencari jenis task pada kata. Pencarian dilakukan dengan string matching KMP. Jika ada, akan menampilkan semua task N hari ke depan. Jika tidak ada, maka lanjut ke langkah selanjutnya.
  - Program akan mencari kata ‘hari ini’ pada kata yang dimasukkan. Jika ada kata yang cocok, maka akan mencari jenis task pada kata. Jika ada, akan menampilkan semua task hari ini. Semua pencarian dilakukan dengan string matching KMP

3. Menampilkan deadline dari suatu task tertentu

- Pertama program akan mengambil kata kunci yang memiliki tipe ‘Task’. Pada database terdapat pada kolom keyword pada relasi new\_task\_keyword dengan kolom type = ‘Task’.
- Lalu program akan mencari kata ‘Tugas <kode matkul>’ pada kata. Pencarian dilakukan dengan menggunakan regex. Jika ada, maka akan mencari kata ‘deadline’ pada kata. Pencarian dilakukan dengan string matching KMP. Jika ada, maka akan menampilkan tanggal dari tugas yang cocok.

4. Memperbarui task tertentu.

- Pertama program akan mengambil kata kunci memperbarui task tertentu. Pada database, terdapat pada kolom keyword pada relasi task\_date\_changed\_keywords.
  - Program akan mencari kata ‘task N’ atau ‘tugas N’ pada kata dengan N adalah integer. Jika ada, maka akan mencari format tanggal. Jika ada, maka akan mengubah deadline task dengan id N menjadi tanggal yang baru. Semua pencarian dilakukan dengan regex.
5. Menandai bahwa suatu task sudah selesai dikerjakan
- Pertama program akan mengambil kata kunci task selesai dikerjakan. Pada database, terdapat pada kolom keyword pada relasi task\_done\_keyword
  - Program akan mencari kata ‘task N’ atau ‘tugas N’ pada kata dengan N adalah integer. Pencarian dilakukan dengan regex. Jika ada, lanjutkan ke langkah berikutnya.
  - Program akan mencari setiap kata kunci bahwa task selesai dikerjakan. Jika ada, maka akan menghapus task dengan id N.

### 3.2 Fitur Fungsional dan Arsitektur *Chatbot* yang Dibangun

1. Chatbot memiliki fitur fungsional berikut :
  - a. Fitur tambah kelas
  - b. Fitur Melihat Daftar Task
  - c. Fitur Menampilkan Deadline suatu Task
  - d. Fitur Memperbarui Deadline suatu Task
  - e. Fitur Menandai suatu Task sudah Selesai Dikerjakan
  - f. Fitur Menampilkan Opsi Bantuan (Help)
2. Chatbot yang dibangun memiliki struktur kode seperti berikut :

```
<?php
include('database.inc.php');

$new_task_keywords = array();
$task_done_keywords = array();
$task_date_changed_keywords = array();
$Tugas_keywords = array();

$sql = "select keyword from `new_task_keywords`";
$res = mysqli_query($con, $sql);
while ($row = mysqli_fetch_assoc($res))
{
    array_push($new_task_keywords, $row['keyword']);
}

$sql = "select keyword from `task_done_keywords`";
$res= mysqli_query($con, $sql);
while ($row = mysqli_fetch_assoc($res))
{
    array_push($task_done_keywords, $row['keyword']);
}

$sql = "select keyword from `task_date_changed_keywords`";
```

```

$res= mysqli_query($con, $sql);
while ($row = mysqli_fetch_assoc($res))
{
    array_push($task_date_changed_keywords, $row['keyword']);
}

$sql = "select keyword from `new_task_keywords` where type = 'Task'";
$res= mysqli_query($con, $sql);
while ($row = mysqli_fetch_assoc($res))
{
    array_push($tugas_keywords, $row['keyword']);
}

$txt = mysqli_real_escape_string($con,$_POST['txt']);

if (newTask($txt, $new_task_keywords) != -1)
{
    makeNewTask($con, $txt, $new_task_keywords);
}
else if (checkAllTaskWithType($txt, $new_task_keywords) != -1)
{
    printAllTask($con, checkAllTaskWithType($txt, $new_task_keywords));
}
else if (checkBetweenDateTaskWithType($txt, $new_task_keywords) != -1)
{
    printTaskBetweenDates($con, $txt, checkBetweenDateTaskWithType($txt, $new_task_keywords));
}
else if (checkNWeekTaskWithType($txt, $new_task_keywords) != -1)
{
    printNWeekTask($con, $txt, checkNWeekTaskWithType($txt, $new_task_keywords));
}
else if (checkNDayTaskWithType($txt, $new_task_keywords) != -1)
{
    printNDayTask($con, $txt, checkNDayTaskWithType($txt, $new_task_keywords));
}
else if (checkTodayTaskWithType($txt, $new_task_keywords) != -1)
{
    printTodayTask($con, checkTodayTaskWithType($txt, $new_task_keywords));
}
else if (checkAllTask($txt) != -1)
{
    printAllTask($con, NULL);
}
else if (checkBetweenDateTask($txt) != -1)
{
    printTaskBetweenDates($con, $txt, NULL);
}
else if (checkNWeekTask($txt) != -1)
{
    printNWeekTask($con, $txt, NULL);
}
else if (checkNDayTask($txt) != -1)
{

```

```

{
    printNDayTask($con, $txt, NULL);
}
else if (checkTodayTask($txt) != -1)
{
    printTodayTask($con, NULL);
}
else if (checkCourseCodeTask($txt) != -1)
{
    printCourseCodeTask($con, $txt, $tugas_keywords);
}
else if (isDoneTask($txt, $task_done_keywords) != -1)
{
    doneTask($con, $txt, $task_done_keywords);
}
else if (isChageDateTask($txt, $task_date_changed_keywords) != -1)
{
    changeDateTask($con, $txt, $task_date_changed_keywords);
}
else if (viewHelpOption($txt) != -1)
{
    echo "
<br>[FITUR & CARA PENGGUNAANNYA]<br>
<b>1. Menambahkan Task Baru<br></b>
Kata kunci: <i>(jenis task) (kode matkul) (topik) pada (tanggal)</i><br>
Format kode matkul = 2 huruf diikuti 4 angka, tanpa spasi<br>
Format tanggal = dd/mm/yyyy<br>
e.g. 'Bot, tambahkan tubes IF2211 String Matching pada 28/04/2021'<br><br>

<b>2. Melihat Daftar Task<br></b>
Contoh kata kunci yang dapat digunakan:<br>


- 'Apa saja deadline yang dimiliki sejauh ini?'<br>
- 'Apa saja deadline antara 01/05/2021 sampai 10/05/2021?'<br>
- 'Deadline 2 minggu ke depan apa saja?'<br>
- 'Deadline 7 hari ke depan apa saja?'<br>
- 'Apa saja deadline hari ini?'<br>
- '3 minggu ke depan ada kuis apa saja?'<br><br>



<b>3. Menampilkan Deadline suatu Task<br></b>
Hanya berlaku untuk task yang bersifat tugas atau memiliki tenggat
waktu.<br>
Contoh kata kunci yang dapat digunakan:<br>


- 'Deadline tugas IF2211 itu kapan'?<br><br>



<b>4. Memperbarui Deadline suatu Task<br></b>
Contoh kata kunci yang dapat digunakan:<br>


- 'Deadline task 12 diundur menjadi 20/05/2021'<br>
- 'Deadline task 20 maju menjadi 01/05/2021'<br><br>



<b>5. Menandai suatu Task sudah Selesai Dikerjakan<br></b>
Contoh kata kunci yang dapat digunakan:<br>


- 'Saya sudah selesai mengerjakan task 16'<br>
- 'Task 20 telah berhasil'<br><br>

```

```

<b>6. Menampilkan Opsi Bantuan (<i>Help</i>)<br></b>
Berisikan command-command yang dapat digunakan oleh pengguna.
Contoh kata kunci yang dapat digunakan:<br>
• 'Apa yang bisa asisten lakukan?'<br>
• 'Tampilkan opsi bantuan'<br><br>

[DAFTAR KATA PENTING]<br>
1. asisten<br>
2. assistant<br>
3. bantuan<br>
4. berhasil<br>
5. bot<br>
6. deadline<br>
7. ditunda<br>
8. diundur<br>
9. done<br>
10. hari ini<br>
11. help<br>
12. kuis<br>
13. lakukan<br>
14. maju<br>
15. mundur<br>
16. N hari ke depan<br>
17. N minggu ke depan<br>
18. praktikum<br>
19. saat ini<br>
20. sejauh ini<br>
21. selesai<br>
22. task<br>
23. tubes<br>
24. tucil<br>
25. tugas<br>
26. ujian<br><br>
";
}
else
{
    echo "Maaf, saya tidak mengerti maksud Anda.<br>";
}

function checkAllTask($text)
{
    if (KMP("Deadline", $text) != -1 || KMP("Task", $text) != -1)
    {
        if (KMP("Sejauh ini", $text) != -1 || KMP("Saat ini", $text) != -1)
        {
            return 1;
        }
    }
    return -1;
}

```

```

function checkAllTaskWithType($text, $new_task_keywords)
{
    if (KMP("Sejauh ini", $text) != -1 || KMP("Saat ini", $text) != -1)
    {
        for ($i = 0; $i < count($new_task_keywords); $i++)
        {
            if (KMP($new_task_keywords[$i], $text) != -1)
            {
                return $new_task_keywords[$i];
            }
        }
    }
    return -1;
}

function printAllTask($con, $task_keyword)
{
    if ($task_keyword == NULL)
    {
        $query = "select * from tasks";
    }
    else
    {
        $query = "select * from tasks where type = '$task_keyword'";
    }

    $res = mysqli_query($con, $query);

    if (mysqli_num_rows($res) == 0) {
        if ($task_keyword == NULL)
        {
            echo "Tidak ada deadline<br>";
        }
        else
        {
            $lowercase_task_keyword = strtolower($task_keyword);
            echo "Tidak ada $lowercase_task_keyword<br>";
        }
    }
    else
    {
        echo "[Daftar Deadline]<br>";
        while ($row = mysqli_fetch_assoc($res))
        {
            echo "(ID: ". $row['id']. ") ". DateTime::createFromFormat('Y-m-d', $row['deadline'])->format('d/m/Y'). " - ". $row['course_id']. " - ". $row['type']. " - ". $row['topic']. "<br>";
        }
    }
}

function checkTodayTask($text)
{

```

```

if (KMP("Deadline", $text) != -1 || KMP("Task", $text) != -1)
{
    if (KMP("Hari ini", $text) != -1)
    {
        return 1;
    }
}
return -1;
}

function checkTodayTaskWithType($text, $new_task_keywords)
{
    if (KMP("Hari ini", $text) != -1)
    {
        for ($i = 0; $i < count($new_task_keywords); $i++)
        {
            if (KMP($new_task_keywords[$i], $text) != -1)
            {
                return $new_task_keywords[$i];
            }
        }
    }
    return -1;
}

function printTodayTask($con, $task_keyword)
{
    $date_now = date('Y-m-d');

    if ($task_keyword == NULL)
    {
        $query = "select * from tasks where deadline = '$date_now'";
    }
    else
    {
        $query = "select * from tasks where deadline = '$date_now' and type = '$task_keyword'";
    }

    $res = mysqli_query($con, $query);

    if (mysqli_num_rows($res) == 0) {
        if ($task_keyword == NULL)
        {
            echo "Tidak ada deadline untuk hari ini<br>";
        }
        else
        {
            $lowercase_task_keyword = strtolower($task_keyword);
            echo "Tidak ada $lowercase_task_keyword untuk hari ini<br>";
        }
    }
    else

```

```

{
    echo "[Daftar Deadline]<br>";
    while ($row = mysqli_fetch_assoc($res))
    {
        echo "(ID: ". $row['id']. " ". DateTime::createFromFormat('Y-m-d',
$row['deadline'])->format('d/m/Y'). " - ". $row['course_id']. " - ". $row['type'].
" - ". $row['topic']). "<br>";
    }
}

function checkBetweenDateTask($text)
{
    if (KMP("Deadline", $text) != -1 || KMP("Task", $text) != -1)
    {
        $date_pattern = "/\d{2}\/\d{2}\/\d{4}/i";
        if (preg_match_all($date_pattern, $text, $matches))
        {
            if (count($matches[0]) == 2)
            {
                return 1;
            }
        }
    }
    return -1;
}

function checkBetweenDateTaskWithType($text, $new_task_keywords)
{
    $date_pattern = "/\d{2}\/\d{2}\/\d{4}/i";
    if (preg_match_all($date_pattern, $text, $matches))
    {
        if (count($matches[0]) == 2)
        {
            for ($i = 0; $i < count($new_task_keywords); $i++)
            {
                if (KMP($new_task_keywords[$i], $text) != -1)
                {
                    return $new_task_keywords[$i];
                }
            }
        }
    }
    return -1;
}

function printTaskBetweenDates($con, $text, $task_keyword)
{
    $date_pattern = "/\d{2}\/\d{2}\/\d{4}/i";
    preg_match_all($date_pattern, $text, $matches);

    $date1 = DateTime::createFromFormat('d/m/Y', $matches[0][0])->format('Y-m-d');
    $date2 = DateTime::createFromFormat('d/m/Y', $matches[0][1])->format('Y-m-d');
}

```

```

if ($task_keyword == NULL)
{
    $query = "select * from tasks where deadline >= '$date1' and deadline <=
'$date2'";
}
else
{
    $query = "select * from tasks where deadline >= '$date1' and deadline <=
$date2' and type = '$task_keyword'";
}

$res = mysqli_query($con, $query);

if (mysqli_num_rows($res) == 0) {
    if ($task_keyword == NULL)
    {
        echo "Tidak ada deadline di antara tanggal tersebut<br>";
    }
    else
    {
        $lowercase_task_keyword = strtolower($task_keyword);
        echo "Tidak ada $lowercase_task_keyword di antara tanggal
tersebut<br>";
    }
}
else
{
    echo "[Daftar Deadline]<br>";
    while ($row = mysqli_fetch_assoc($res))
    {
        echo "(ID: ". $row['id']. ") ". DateTime::createFromFormat('Y-m-d',
$row['deadline'])->format('d/m/Y'). " - ". $row['course_id']. " - ". $row['type']. "
- ". $row['topic']. "<br>";
    }
}
}

function checkNDayTask($text)
{
    if (KMP("Deadline", $text) != -1 || KMP("Task", $text) != -1)
    {
        $day_pattern = "/(\d+)\s*hari ke depan/i";
        if (preg_match($day_pattern, $text, $matches))
        {
            return 1;
        }
    }
    return -1;
}

function checkNDayTaskWithType($text, $new_task_keywords)
{

```

```

$day_pattern = "/(\d+)\s*hari ke depan/i";
if (preg_match($day_pattern, $text, $matches))
{
    for ($i = 0; $i < count($new_task_keywords); $i++)
    {
        if (KMP($new_task_keywords[$i], $text) != -1)
        {
            return $new_task_keywords[$i];
        }
    }
}
return -1;
}

function printNDayTask($con, $text, $task_keyword)
{
    $day_pattern = "/(\d+)\s*hari ke depan/i";
    preg_match($day_pattern, $text, $matches);
    $day_count_pattern = "/(\d+)/i";
    preg_match($day_count_pattern, $matches[0], $day_count);

    $date_now = date('Y-m-d');
    $date_now = new DateTimeImmutable($date_now);
    $date_Nday = $date_now->modify("+{$day_count[0]} day");

    $date_now_str = $date_now->format('Y-m-d');
    $date_Nday_str = $date_Nday->format('Y-m-d');

    if ($task_keyword == NULL)
    {
        $query = "select * from tasks where deadline >= '$date_now_str' and
deadline <= '$date_Nday_str'";
    }
    else
    {
        $query = "select * from tasks where deadline >= '$date_now_str' and
deadline <= '$date_Nday_str' and type = '$task_keyword'";
    }

    $res = mysqli_query($con, $query);

    if (mysqli_num_rows($res) == 0) {
        if ($task_keyword == NULL)
        {
            echo "Tidak ada deadline {$day_count[0]} hari ke depan<br>";
        }
        else
        {
            $lowercase_task_keyword = strtolower($task_keyword);
            echo "Tidak ada $lowercase_task_keyword {$day_count[0]} hari ke
depan<br>";
        }
    }
}

```

```

        }
    else
    {
        echo "[Daftar Deadline]<br>";
        while ($row = mysqli_fetch_assoc($res))
        {
            echo "(ID: ". $row['id']. " ". DateTime::createFromFormat('Y-m-d',
$row['deadline'])->format('d/m/Y'). " - ". $row['course_id']. " - ". $row['type'].
" - ". $row['topic']. "<br>";
        }
    }
}

function checkNWeekTask($text)
{
    if (KMP("Deadline", $text) != -1 || KMP("Task", $text) != -1)
    {
        $day_pattern = "/(\d+)\s*minggu ke depan/i";
        if (preg_match($day_pattern, $text, $matches))
        {
            return 1;
        }
    }
    return -1;
}

function checkNWeekTaskWithType($text, $new_task_keywords)
{
    $day_pattern = "/(\d+)\s*minggu ke depan/i";
    if (preg_match($day_pattern, $text, $matches))
    {
        for ($i = 0; $i < count($new_task_keywords); $i++)
        {
            if (KMP($new_task_keywords[$i], $text) != -1)
            {
                return $new_task_keywords[$i];
            }
        }
    }
    return -1;
}

function printNWeekTask($con, $text, $task_keyword)
{
    $week_pattern = "/(\d+)\s*minggu ke depan/i";
    preg_match($week_pattern, $text, $matches);
    $week_count_pattern = "/(\d+)/i";
    preg_match($week_count_pattern, $matches[0], $week_count);

    $date_now = date('Y-m-d');
    $date_now = new DateTimeImmutable($date_now);
    $date_Nweek = $date_now->modify("+$week_count[0] week");
}

```

```

$date_now_str = $date_now->format('Y-m-d');
$date_Nweek_str = $date_Nweek->format('Y-m-d');

if ($task_keyword == NULL)
{
    $query = "select * from tasks where deadline >= '$date_now_str' and
deadline <= '$date_Nweek_str'";
}
else
{
    $query = "select * from tasks where deadline >= '$date_now_str' and
deadline <= '$date_Nweek_str' and type = '$task_keyword'";
}

$res = mysqli_query($con, $query);

if (mysqli_num_rows($res) == 0) {
    if ($task_keyword == NULL)
    {
        echo "Tidak ada deadline $week_count[0] minggu ke depan<br>";
    }
    else
    {
        $lowercase_task_keyword = strtolower($task_keyword);
        echo "Tidak ada $lowercase_task_keyword $week_count[0] minggu ke
depan<br>";
    }
}
else
{
    echo "[Daftar Deadline]<br>";
    while ($row = mysqli_fetch_assoc($res))
    {
        echo "(ID: ". $row['id']. ") ". DateTime::createFromFormat('Y-m-d',
$row['deadline'])->format('d/m/Y'). " - ". $row['course_id']. " - ". $row['type']. "
- ". $row['topic']. "<br>";
    }
}
}

function checkCourseCodeTask($text)
{
    $course_code_pattern = "/Tugas\s*[A-Za-z]{2}\d{4}\s*/i";
    if (!preg_match($course_code_pattern, $text, $matches))
    {
        return - 1;
    }

    if (KMP("Deadline", $text) != -1)
    {
        return trim(substr($matches[0], 5));
    }
}

```

```

        return -1;
    }

    function printCourseCodeTask($con, $text, $tugas_keywords)
    {
        $date_now = date('Y-m-d');
        $course_code = checkCourseCodeTask($text);
        if (count($tugas_keywords) == 1)
        {
            $query = "select * from tasks where type = '$tugas_keywords[0]'";
        }
        else
        {
            $query = "select * from tasks where (type = '$tugas_keywords[0]' ";
        }

        for ($i = 1; $i < count($tugas_keywords) - 1; $i++)
        {
            $query = $query . " or type = '$tugas_keywords[$i]' ";
        }

        $last_index = count($tugas_keywords) - 1;
        if (count($tugas_keywords) > 1)
        {
            $query = $query . " or type = '$tugas_keywords[$last_index]' ";
        }

        $query = $query . " and LOWER(course_id) = LOWER('$course_code') and deadline >
        '$date_now'";

        $res = mysqli_query($con, $query);

        if (mysqli_num_rows($res) == 0) {
            echo "Tidak ada deadline dari mata kuliah tersebut<br>";
        }
        else
        {
            while ($row = mysqli_fetch_assoc($res))
            {
                echo DateTime::createFromFormat('Y-m-d',
$row['deadline'])->format('d/m/Y') . "<br>";
            }
        }
    }

    function newTask($text, $new_task_keywords)
    {
        for ($i = 0; $i < count($new_task_keywords); $i++)
        {
            $pattern      =
"/$new_task_keywords[$i]\s*[A-Za-z]{2}\d{4}\s*(.*)\s*pada\s*\d{2}\/\d{2}\/\d{4}/i";
            if (preg_match($pattern, $text, $matches, PREG_OFFSET_CAPTURE))
            {
                return $matches[0][1];
            }
        }
    }
}

```

```

        }

        return -1;
    }

function makeNewTask($con, $text, $new_task_keywords)
{
    $task_str = substr($text, newTask($text, $new_task_keywords));
    for ($i = 0; $i < count($new_task_keywords); $i++)
    {
        if (KMP($new_task_keywords[$i], $task_str) != -1)
        {
            $task_type = $new_task_keywords[$i];
            break;
        }
    }

$course_code_pattern = "/[A-Za-z]{2}\d{4}/i";
preg_match($course_code_pattern, $task_str, $course_code, PREG_OFFSET_CAPTURE);

$pattern = "/pada\s*\d{2}\.\d{2}\.\d{4}/i";
preg_match($pattern, $task_str, $date_start, PREG_OFFSET_CAPTURE);
$topic = substr($task_str, $course_code[0][1] + 6, $date_start[0][1] - 1 - ($course_code[0][1] + 6));

$deadline = substr($date_start[0][0], 5);
$deadline = trim($deadline);
$deadline = DateTime::createFromFormat('d/m/Y', $deadline);

insertTaskDB($con, $course_code[0][0], $task_type, $topic, $deadline);
}

function isChageDateTask($text, $task_date_changed_keywords)
{
    $pattern = "/Task\s*(\d+)\s*/i";
    if (!preg_match($pattern, $text, $matches))
    {
        $pattern = "/Tugas\s*(\d+)\s*/i";
        if (!preg_match($pattern, $text, $matches))
        {
            return -1;
        }
    }

$date = "/\d{2}\.\d{2}\.\d{4}/i";
if (!preg_match($date, $text, $matches))
{
    return -1;
}

for ($i = 0; $i < count($task_date_changed_keywords); $i++)
{
    if (KMP($task_date_changed_keywords[$i], $text) != -1)
    {
        return 1;
    }
}

```

```

        }
    }
    return -1;
}

function changeDateTask($con, $text, $task_date_changed_keywords)
{
    $task_id_pattern = "/Task\s*(\d+)\s*/i";
    if (preg_match($task_id_pattern, $text, $matches))
    {
        $task_id = substr($matches[0], 4);
    }

    $task_id_pattern = "/Tugas\s*(\d+)\s*/i";
    if (preg_match($task_id_pattern, $text, $matches))
    {
        $task_id = substr($matches[0], 5);
    }

    $date_pattern = "/\d{2}\/\d{2}\/\d{4}/i";
    preg_match($date_pattern, $text, $matches);
    $date = DateTime::createFromFormat('d/m/Y', $matches[0]);

    renewTaskDB($con, $task_id, $date);
}

function isDoneTask($text, $task_done_keywords)
{
    $pattern = "/Task\s*(\d+)\s*/i";
    if (!preg_match($pattern, $text, $matches))
    {
        $pattern = "/Tugas\s*(\d+)\s*/i";
        if (!preg_match($pattern, $text, $matches))
        {
            return -1;
        }
        else
        {
            $task_id = substr($matches[0], 5);
        }
    }
    else
    {
        $task_id = substr($matches[0], 4);
    }
    for ($i = 0; $i < count($task_done_keywords); $i++)
    {
        if (KMP($task_done_keywords[$i], $text) != -1)
        {
            return $task_id;
        }
    }
    return -1;
}

function doneTask($con, $text, $new_task_keywords)

```

```

{
    $id = isDoneTask($text, $new_task_keywords);
    deleteTaskDB($con, $id);
}

function insertTaskDB($con, $course_code, $type, $topic, $date)
{
    $strdate = $date->format('Y-m-d');
    $query = "insert into tasks (course_id, type, deadline, topic) values
    ('$course_code', '$type', '$strdate', '$topic');";
    mysqli_query($con, $query);
    if (mysqli_affected_rows($con) > 0)
    {
        $res = mysqli_query($con, "select * from tasks order by id desc limit 1");
        $row = mysqli_fetch_assoc($res);
        echo "[TASK BERHASIL DICATAT]<br>". "(ID: ". $row['id']. " ) ".
        DateTime::createFromFormat('Y-m-d', $row['deadline'])->format('d/m/Y'). " - ".
        $row['course_id']. " - ". $row['type']. " - ". $row['topic'];
    }
    else
    {
        echo "Error: " . $query . "<br>" . mysqli_error($con);
    }
}

function renewTaskDB($con, $id, $new_date)
{
    $strdate = $new_date->format('Y-m-d');
    $query = "update tasks set deadline = '$strdate' where id = '$id'";
    mysqli_query($con, $query);
    if (mysqli_affected_rows($con) > 0)
    {
        echo "Task". $id. " diupdate";
    }
    else
    {
        echo "Task belum terdaftar";
    }
}

function deleteTaskDB($con, $id)
{
    $query = "delete from tasks where id = '$id'";
    mysqli_query($con, $query);
    if (mysqli_affected_rows($con) > 0)
    {
        echo "Task". $id. " ditandai selesai";
    }
    else
    {
        echo "Task belum terdaftar";
    }
}

function viewHelpOption($text)
{
}

```

```

    if (KMP("Assistant", $text) != -1 || KMP("Asister", $text) != -1 || KMP("Bot",
$text) != -1)
    {
        if (KMP("Lakukan", $text) != -1)
        {
            return 1;
        }
    }
    else if (KMP("Bantuan", $text) != -1 || KMP("Help", $text) != 1)
    {
        return 1;
    }
    return -1;
}

function KMP($pattern, $text)
{
    $M = strlen($pattern);
    $N = strlen($text);
    $lps = LPSarray($pattern);
    $i = 0;
    $j = 0;
    while ($i < $N) {
        if (strtolower($pattern[$j]) == strtolower($text[$i]))
        {
            $j++;
            $i++;
        }
        if ($j == $M)
        {
            $idx = $i - $j;
            return $idx;
        }
        else if ($i < $N && strtolower($pattern[$j]) != strtolower($text[$i]))
        {
            if ($j == 0)
            {
                $i = $i + 1;
            }
            else
            {
                $j = $lps[$j - 1];
            }
        }
    }
    return -1;
}

function LPSarray($pattern)
{
    $len = 0;

    $lps = array();
    $lps[0] = 0;
}

```

```
$i = 1;
while ($i < strlen($pattern)) {
    if (strtolower($pattern[$i]) == strtolower($pattern[$len]))
    {
        $len++;
        $lps[$i] = $len;
        $i++;
    }
    else
    {
        if ($len != 0)
        {
            $len = $lps[$len - 1];
        }
        else
        {
            $lps[$i] = 0;
            $i++;
        }
    }
}
return $lps;
}
?>
```

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1 Spesifikasi Teknis Program**

Pada database, terdapat 4 relasi yang terpakai, yaitu new\_task\_keywords, task\_date\_changed\_keywords, task\_done\_keywords, dan tasks. New\_task\_keywords akan menyimpan kata kunci untuk membuat task baru dan tipe dari task, task\_date\_changed\_keywords akan menyimpan kata kunci yang diperlukan untuk mengubah tanggal, task\_done\_keywords akan menyimpan kata kunci yang diperlukan untuk menandai bahwa tugas sudah selesai, task akan berisi seluruh data task.

Pada file get\_bot\_message.php terdapat beberapa fungsi yang dibuat, yaitu:

- checkAllTask(\$text)  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan seluruh task saat ini
- checkAllTaskWithType(\$text, \$new\_task\_keywords)  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan seluruh task saat ini dengan tipe task tertentu
- printAllTask(\$con, \$task\_keyword)  
Digunakan untuk menampilkan seluruh task saat ini. Jika task\_keyword tidak null, maka akan menampilkan seluruh task saat ini dengan tipe task tertentu.
- checkTodayTask(\$text)  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan seluruh task hari ini
- checkTodayTaskWithType(\$text, \$new\_task\_keywords)  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan seluruh task hari ini dengan tipe task tertentu
- printTodayTask(\$con, \$task\_keyword)  
Digunakan untuk menampilkan task pada hari ini. Jika task keyword tidak null, maka akan menampilkan task hari ini dengan tipe task tertentu.
- checkBetweenDateTask(\$text)  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan task yang berada di antara 2 format tanggal
- checkBetweenDateTaskWithType(\$text, \$new\_task\_keywords)  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan task yang berada di antara 2 format tanggal dengan tipe task tertentu
- printTaskBetweenDates(\$con, \$text, \$task\_keyword)  
Digunakan untuk menampilkan task yang berada pada 2 format tanggal. Jika task\_keyword tidak null, maka hanya akan menampilkan task dengan tipe tertentu.
- checkNDayTask(\$text)  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan N hari ke depan
- checkNDayTaskWithType(\$text, \$new\_task\_keywords)

- **printNDayTask(\$con, \$text, \$task\_keyword)**  
Digunakan untuk menampilkan task pada N hari ke depan dengan tipe task tertentu.
- **checkNWeekTask(\$text)**  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan N minggu ke depan
- **checkNWeekTaskWithType(\$text, \$new\_task\_keywords)**  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan N minggu ke depan dengan tipe task tertentu
- **printNWeekTask(\$con, \$text, \$task\_keyword)**  
Digunakan untuk menampilkan task pada N minggu ke depan. Jika task keyword tidak null, akan menampilkan task dengan tipe tertentu
- **checkCourseCodeTask(\$text)**  
Digunakan untuk mengecek jika input user bertujuan untuk menampilkan tanggal deadline dari suatu task yang memiliki ‘tipe’ = task pada new\_task\_keyword
- **printCourseCodeTask(\$con, \$text, \$tugas\_keywords)**  
Digunakan untuk menampilkan tanggal deadline dari suatu task yang memiliki ‘tipe’ = task pada new\_task\_keyword
- **newTask(\$text, \$new\_task\_keywords)**  
Digunakan untuk menambahkan task ke database
- **isChageDateTask(\$text, \$task\_date\_changed\_keywords)**  
Digunakan untuk mengecek jika input user bertujuan untuk mengganti tanggal deadline dari id task tertentu
- **changeDateTask(\$con, \$text, \$task\_date\_changed\_keywords)**  
Digunakan untuk mengganti deadline suatu task dengan id tertentu pada database
- **isDoneTask(\$text, \$task\_done\_keywords)**  
Digunakan untuk mengecek jika input user bertujuan menghapus suatu task
- **doneTask(\$con, \$text, \$new\_task\_keywords)**  
Digunakan untuk menghapus task dengan id tertentu ke database
- **insertTaskDB(\$con, \$course\_code, \$type, \$topic, \$date)**  
Digunakan untuk memasukkan task baru ke database
- **renewTaskDB(\$con, \$id, \$new\_date)**  
Digunakan untuk memperbarui tanggal deadline dengan id task tertentu
- **deleteTaskDB(\$con, \$id)**  
Digunakan untuk menghapus task dengan id tertentu ke database
- **viewHelpOption(\$text)**  
Digunakan untuk mengecek jika user input bertujuan untuk melihat help
- **KMP(\$pattern, \$text)**  
Digunakan untuk string matching dengan metode KMP
- **LPSarray(\$pattern)**  
Utilitas yang digunakan untuk membuat array LPS untuk kebutuhan string matching dengan metode KMP

## 4.2 Penjelasan Tata Cara Penggunaan Program

Aplikasi *chatbot* ini dibangun berbasis web (*web-based application*). Jika ingin menjalankan aplikasi *chatbot* di server lokal:

1. Instal XAMPP terlebih dahulu. Jalankan XAMPP kemudian aktifkan modul Apache dengan mengklik tombol Start.
2. Buat folder baru dengan nama misalkan "chatbot" di direktori C:/xampp/htdocs.
3. Copy isi folder src ke dalam folder chatbot.
4. Buka aplikasi peramban (browser), contoh: Google Chrome.
5. Ketik "localhost/chatbot/index.php" di tab pencarian lalu klik Enter.

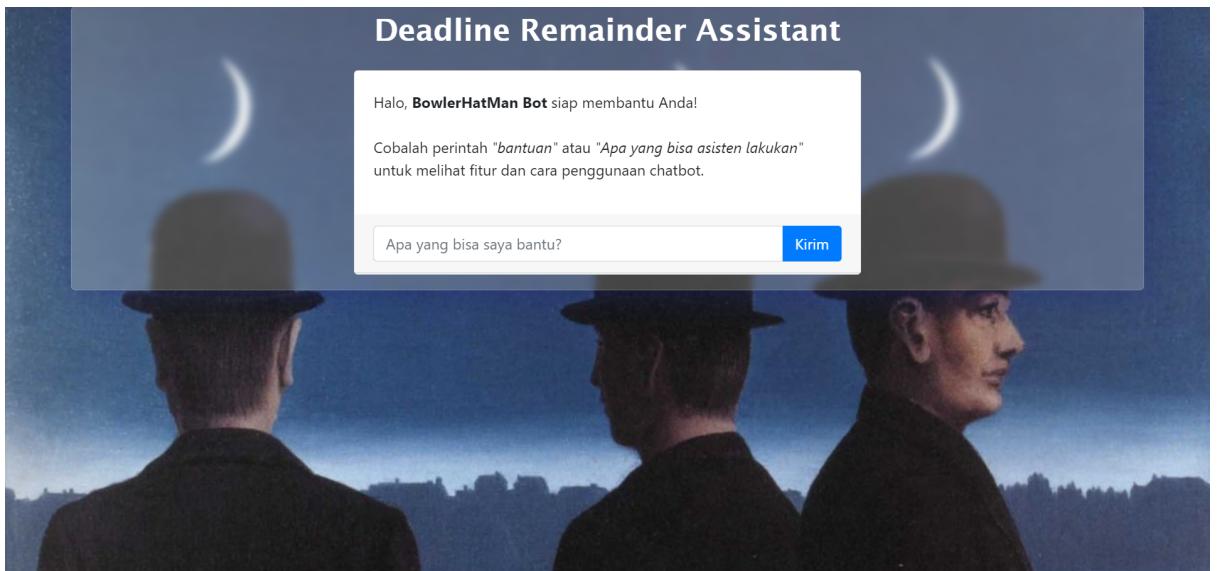
5. Silakan gunakan aplikasi chatbot.

Halaman awal *chatbot* terdiri dari judul dan form untuk memasukkan perintah ke bot yang disertai tombol Kirim untuk mengirimkan perintahnya. Fitur-fitur *chatbot* terdiri dari fitur tambah *task*, fitur melihat daftar *task*, fitur menampilkan deadline suatu *task*, fitur memperbarui deadline suatu *task*, fitur menandai suatu *task* sudah selesai dikerjakan, dan fitur opsi bantuan (*help*).

## 4.3 Hasil Pengujian

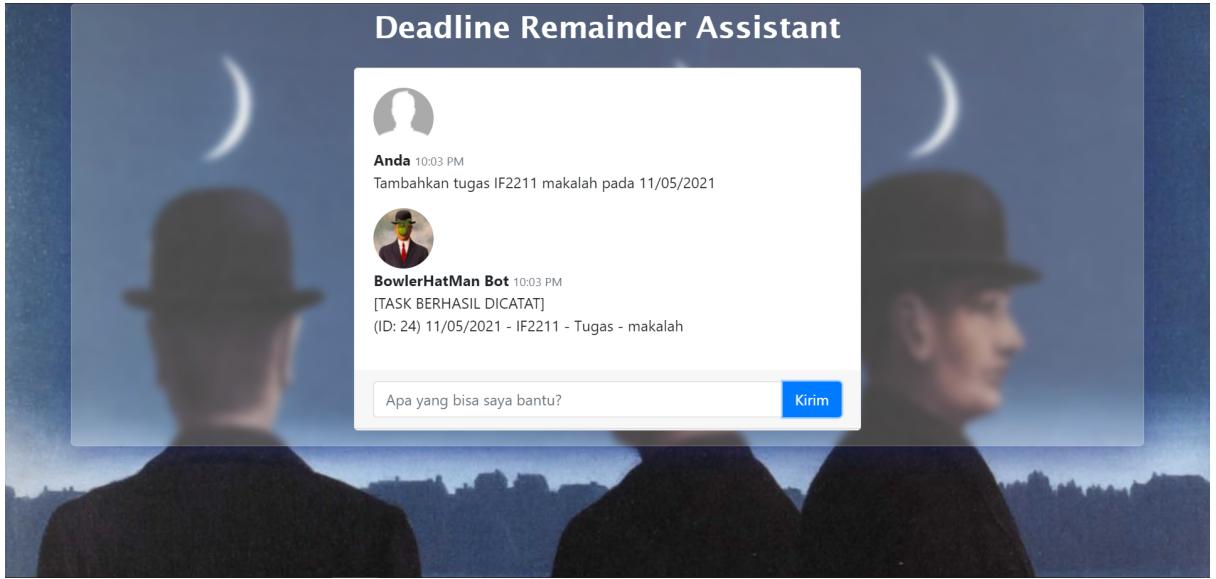
*Screenshot* antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur dalam Chatbot.

### 4.3.1 Halaman Awal



Gambar 4.1 halaman awal web

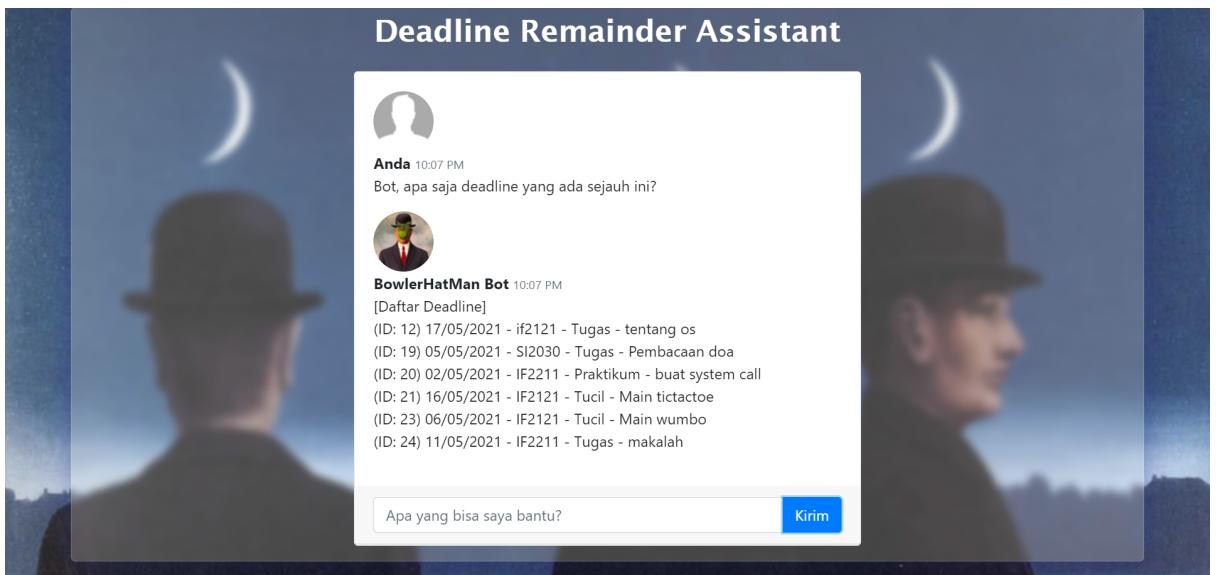
### 4.3.2 Fitur Tambah Task



Gambar 4.2 tambah task dalam web

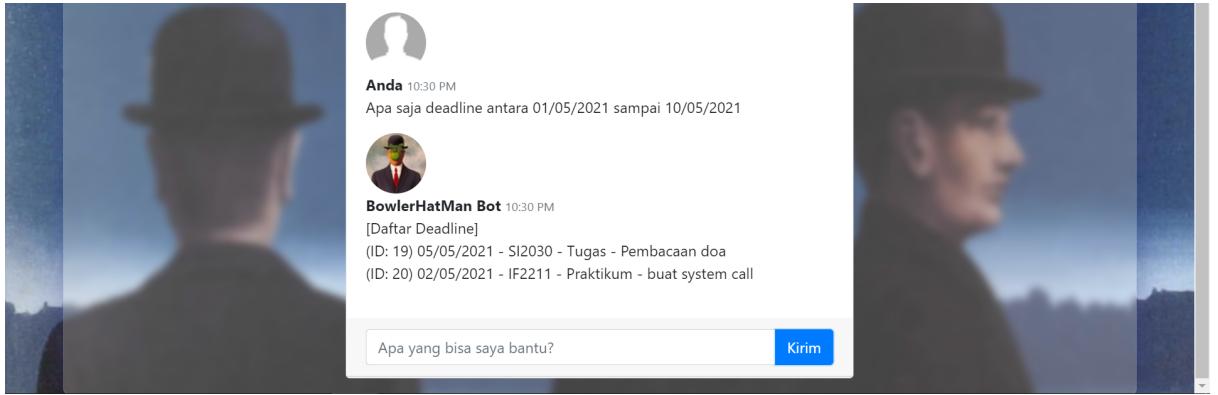
#### 4.3.3 Fitur Melihat Daftar Task

Skenario 1: Seluruh *task* yang sudah tercatat oleh assistant.



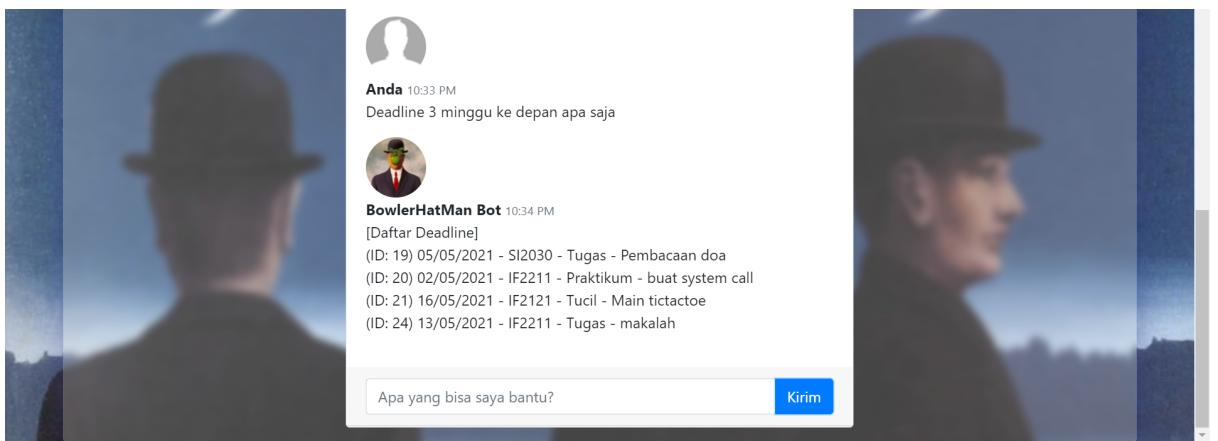
Gambar 4.3.1 lihat task dalam web yang sudah tercatat

Skenario 2: Berdasarkan periode waktu (dari tanggal 1 sampai tanggal 2).



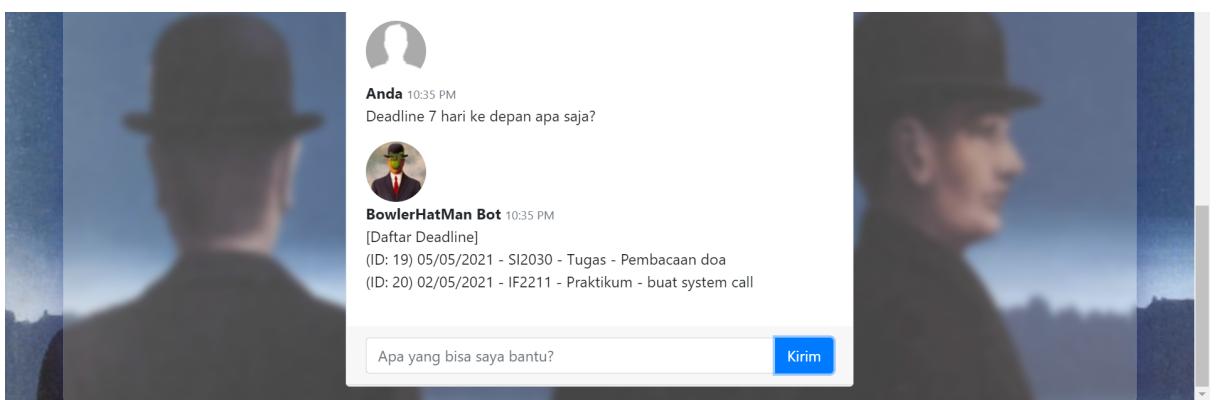
Gambar 4.3.2 lihat task dalam web berdasarkan tanggal tertentu

Skenario 3: Berdasarkan periode waktu (N minggu ke depan).



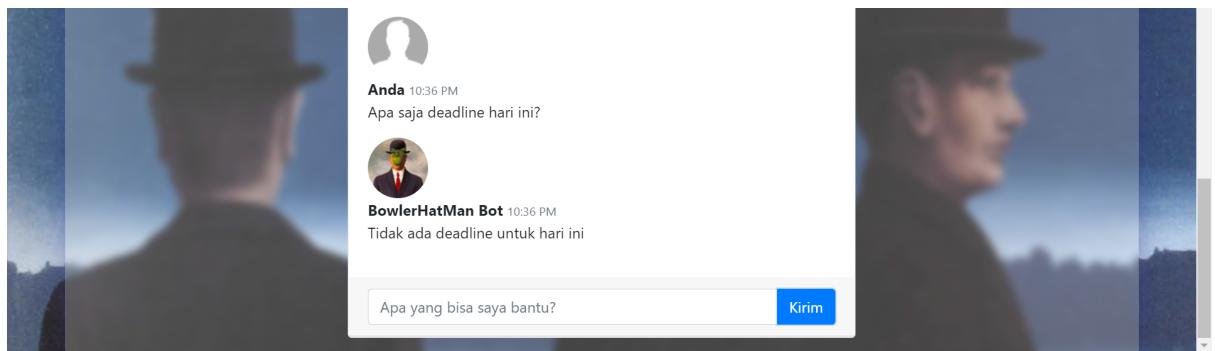
Gambar 4.3.3 lihat task dalam web berdasarkan N minggu kedepan

Skenario 4: Berdasarkan periode waktu (N hari ke depan).



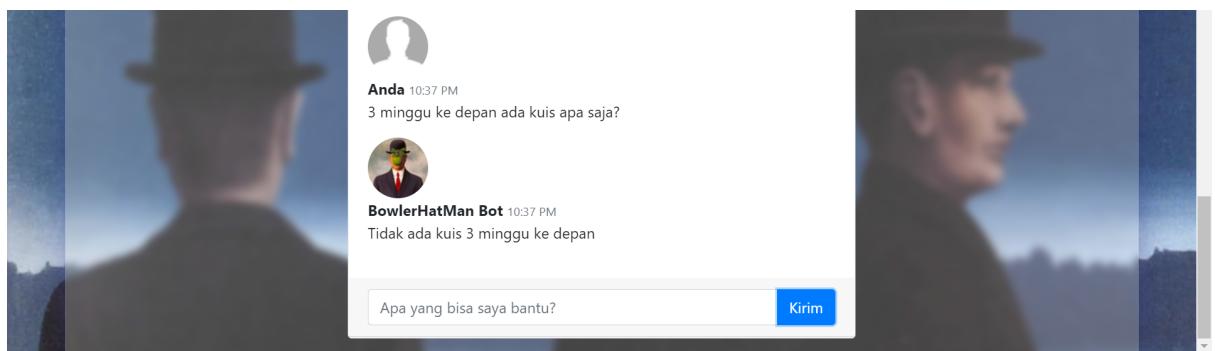
Gambar 4.3.4 lihat task dalam web berdasarkan N hari kedepan

Skenario 5: Skenario 3: Berdasarkan periode waktu (hari ini).



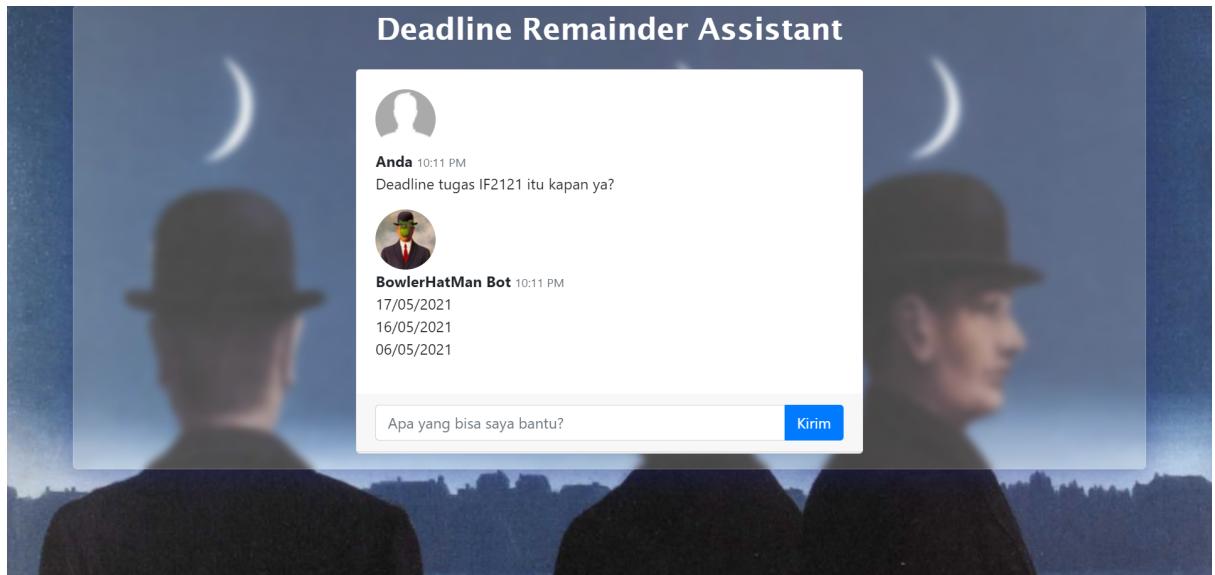
Gambar 4.3.5 lihat task dalam web hari ini

Skenario 6: Berdasarkan jenis *task* (kata penting).



Gambar 4.3.6 lihat task dalam web berdasarkan keyword

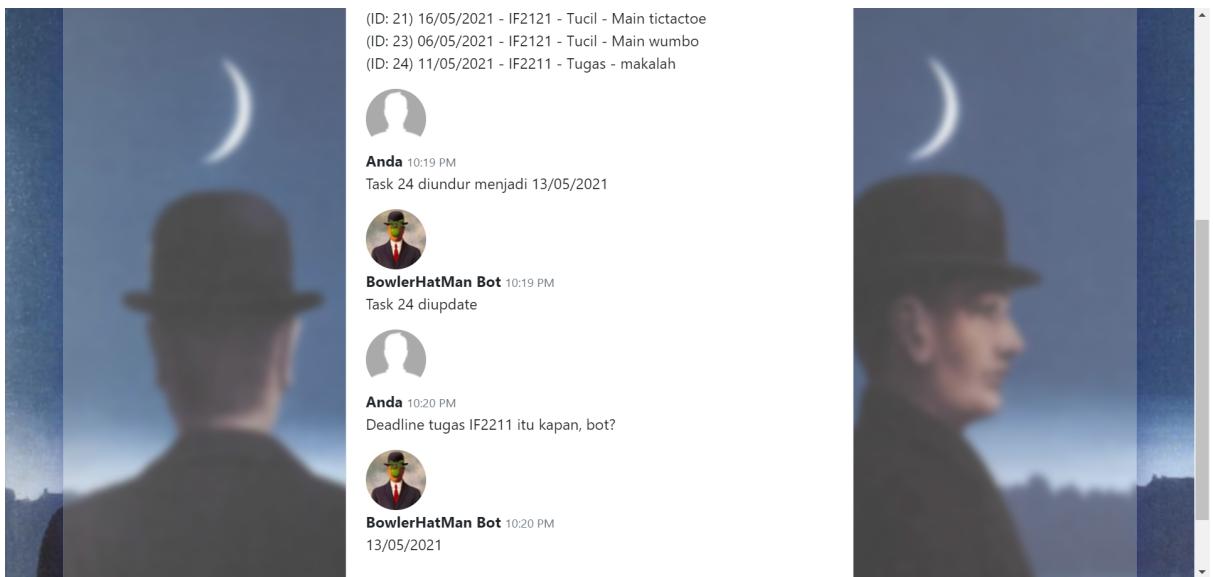
#### 4.3.4 Fitur Menampilkan Deadline suatu Task



Gambar 4.4 Menampilkan deadline task dalam web

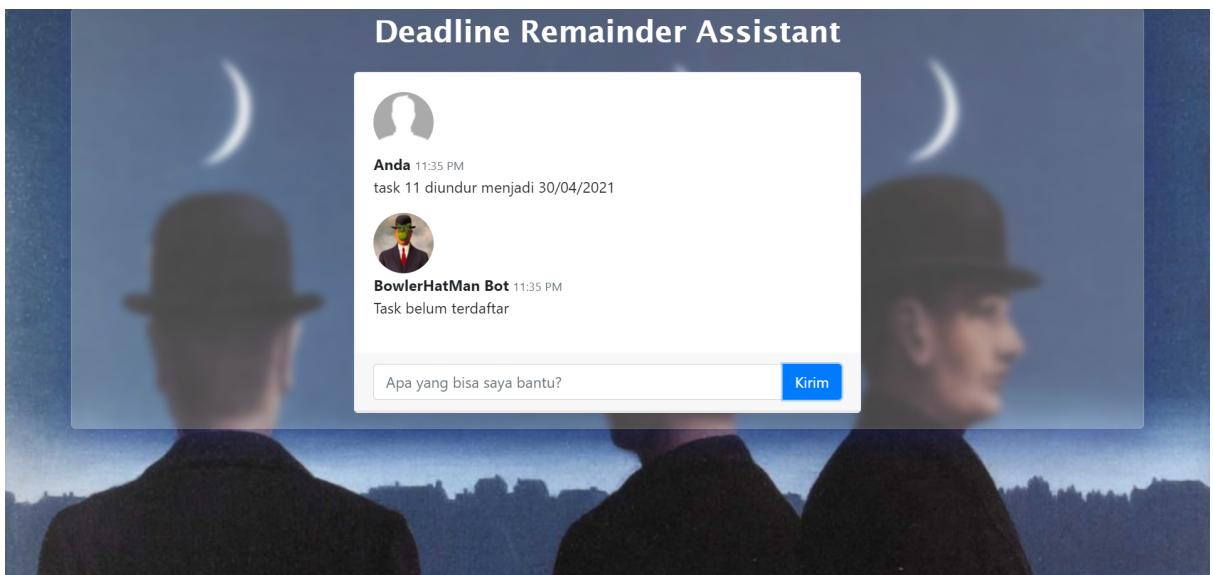
#### 4.3.5 Fitur Memperbarui Deadline suatu Task

Skenario 1: Task yang diperbarui sudah terdaftar.



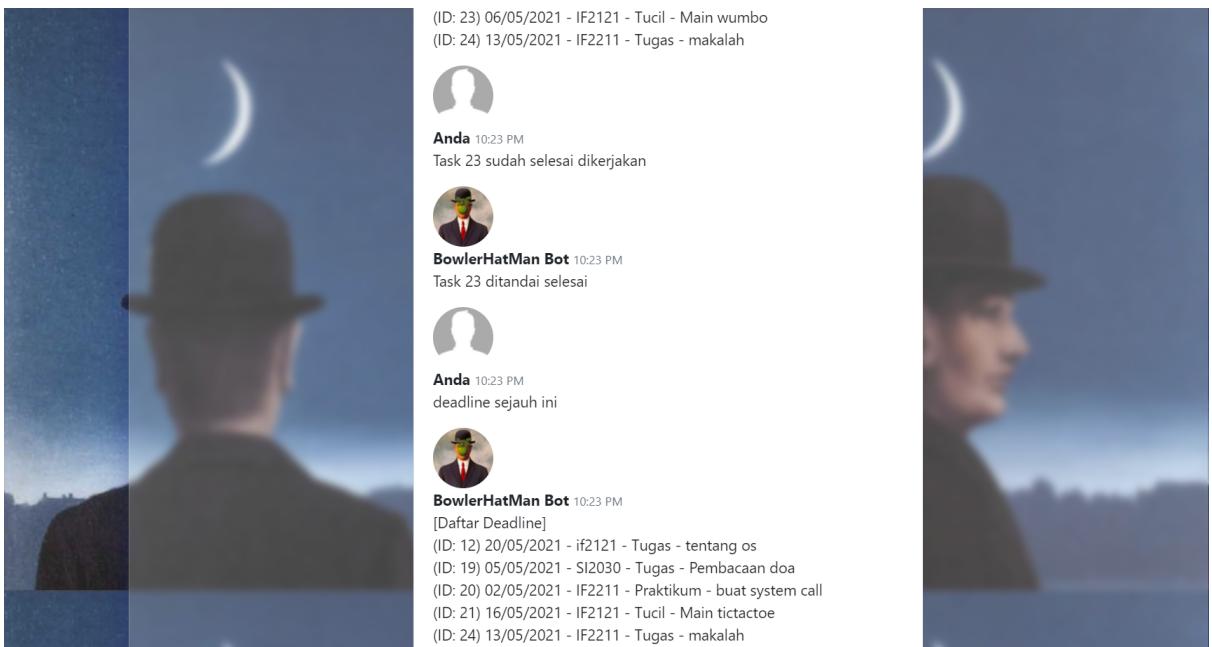
Gambar 4.5.1 Perbaharui deadline task yang sudah terdaftar

Skenario 2: Task yang diperbarui belum terdaftar.



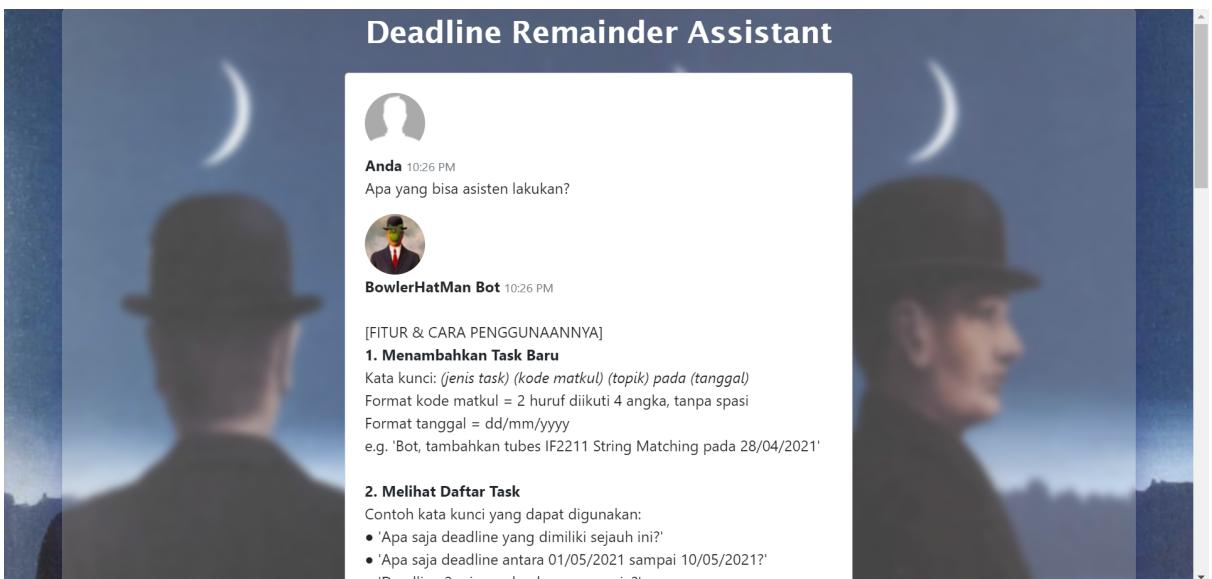
Gambar 4.5.2 Perbaharui deadline task yang tidak terdaftar

#### 4.3.6 Fitur Menandai suatu Task sudah Selesai Dikerjakan



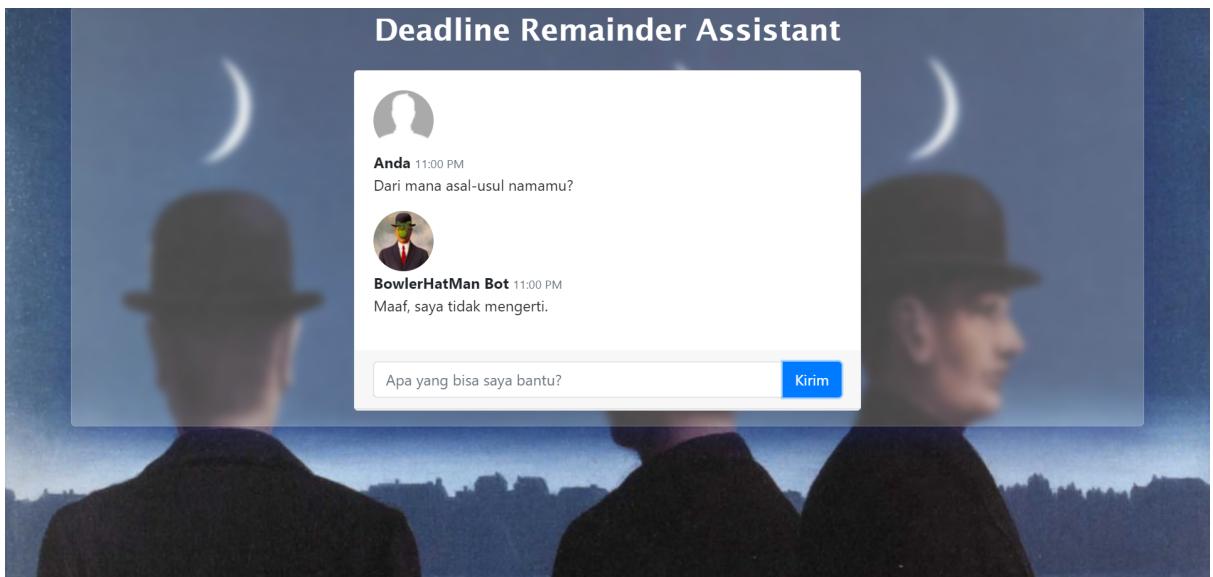
Gambar 4.6 Mark task sudah selesai dalam web

#### 4.3.7 Fitur Menampilkan Opsi Bantuan (*Help*)



Gambar 4.7 Tampilan help dalam web

#### 4.3.8 Merespons Pesan yang Tidak Dikenali



Gambar 4.8 Respon bot dari pertanyaan yang tidak terdapat dalam database web

#### 4.4 Analisis Hasil Pengujian

Poin	Ya	Tidak
Chatbot dapat menerima input string dari user.	✓	
Chatbot dapat membaca input user dan mencari keyword yang tepat dalam database	✓	
Chatbot dapat memberikan tanggapan yang sesuai untuk setiap fitur dalam spek	✓	
Chatbot diimplementasikan secara web based	✓	
Bonus: Chatbot dapat mengoreksi suatu keyword yang typo dan memberikan rekomendasi keyword yang benar		✓

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Chatbot merupakan AI yang dapat mengetahui apa yang sedang dibacanya dengan melakukan pencocokan string kepada text dan mencari keyword paling relevan yang ada pada databasenya. Chatbot diterapkan pada banyak hal, tidak jarang pula ditemui pada aplikasi berbasis web seperti website bisnis, website call center, atau fitur search text yang ada pada IDE maupun notepad, dan lain sebagainya. Karena chatbot menerapkan algoritma pattern matching, biasanya algoritma yang digunakan adalah algoritma Knuth-Morris-Pratt (KMP), atau algoritma Boyer-Moore (BM), atau bisa juga menggunakan Regex. Dalam pembuatan programnya, membuat chatbot paling cocok menggunakan python, java, atau php.

#### **5.2 Saran**

Chatbot yang kami buat merupakan chatbot yang berbasis pada database, dan sudah sesuai dengan spek dari tugas yang diberikan. Namun akan menjadi nilai tambah lagi jika kedepannya kami dapat membuat chatbot yang memiliki fitur deep learning sehingga chatbot akan jauh lebih dinamis dan lebih interaktif seperti beberapa chatbot yang sudah ada selama ini.

#### **5.3 Refleksi dan Komentar terhadap Tugas**

Refleksi dari pembuatan tugas besar ini adalah :

1. Leonardus Brandon Luwianto (13519102)

Kami bangga dapat menyelesaikan tubes ini hanya dalam waktu 2 hari.

2. Made Kharisma Jagaddhita (13519176)

Pengen tidur

3. Farrell Abieza Zidan (13519182)

Tugas besar “deadline reminder assistant” harus dibuat secara deadline pula.

## **DAFTAR PUSTAKA**

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf> diakses pada tanggal 26 April 2021.
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf> diakses pada tanggal 26 April 2021.
- [3] <https://www.itechart.com/blog/how-do-chatbots-really-work/#:~:text=It%20is%20designed%20specifically%20to,bot%20become%20more%20human%2Dlike.&text=It%20will%20then%20present%20the,to%20specific%20AI%20chatbot%20algorithms>. Diakses pada tanggal 27 April 2021.
- [4] <https://iq.opengenus.org/kmp-vs-boyer-moore-algorithm/> Diakses pada tanggal 28 April 2021