

Tugas Besar III IF2211 Strategi Algoritma

Semester II Tahun 2020/2021

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



Disusun oleh:

1. Karlсен Adiyasa Bachtiar (13519001)
2. Christian Gunawan (13519199)
3. Muhammad Fawwaz Naabigh (13519206)

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

2021

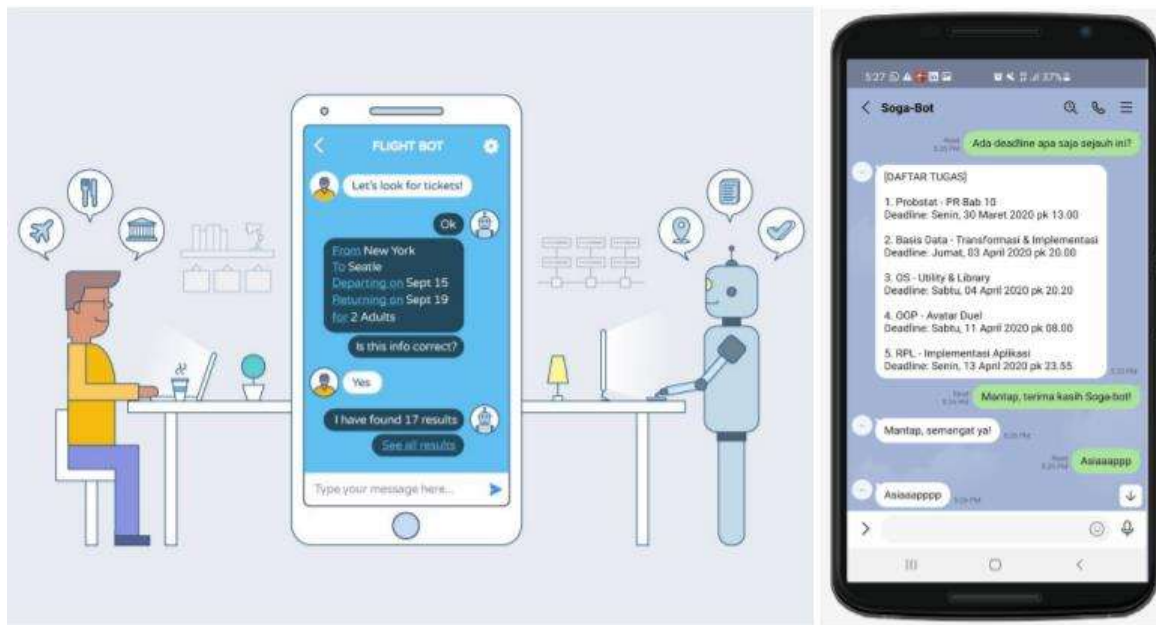
Daftar Isi

Bab 1.....	3
Deskripsi Tugas	3
Bab 2.....	4
Landasan Teori.....	4
2.1 Algoritma Knuth-Morris-Pratt.....	4
2.2 Algoritma Boyer Moore.....	4
2.3 Reguler Expression.....	4
2.4 Penjelasan singkat mengenai Chatbot	4
Analisis Pemecahan Masalah	5
3.1 Langkah penyelesaian masalah setiap fitur	5
3.2 Fitur fungsional dan arsitektur Chatbot yang dibangun.....	5
Bab 4.....	6
Implementasi dan Pengujian.....	6
4.1 Spesifikasi teknis program (struktur data, fungsi, prosedur yang dibangun)	6
4.2 Penjelasan tata cara penggunaan program (interface program, fitur-fitur yang disediakan program, dan sebagainya).....	8
4.3 Hasil pengujian (screenshot antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur dalam Chatbot).....	10
4.4 Analisis hasil pengujian.....	11
Bab 5.....	12
Kesimpulan.....	12
5.1 Kesimpulan.....	12
5.2 Saran	12
5.3 Komentar.....	12
Daftar Pustaka.....	13

Bab 1

Deskripsi Tugas

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



Gambar 1. Ilustrasi Chatbot dan Asisten Pengingat Deadline

Sumber : <https://id.pinterest.com/pin/824299538024636729/>

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.

Bab 2

Landasan Teori

2.1 Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt atau biasa disebut Algoritma KMP merupakan algoritma yang digunakan untuk melakukan proses pencocokan string. Algoritma ini merupakan jenis Exact String Matching Algorithm yang merupakan pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang sama. Algoritma ini dapat memiliki kompleksitas algoritma yang jauh lebih baik dari algoritma *bruteforce*. Algoritma ini memiliki kompleksitas algoritma $O(m+n)$. Algoritma ini lebih baik jika dibandingkan dengan algoritma *bruteforce* karena tidak adanya dilakukan backtracking pada teks.

2.2 Algoritma Boyer Moore

Algoritma Boyer Moore atau biasa disebut Algoritma BM adalah algoritma yang melakukan pencarian kata mulai dari indeks terakhir *Pattern* sampai akhirnya mencapai posisi paling kiri, teknik ini sering dikenal dengan “the looking-glass technique”. Langkah ini berbeda dengan algoritma pencarian string yang serupa (seperti *bruteforce* dan Knuth-Morris-Pratt algorithm) yang memulai pencarian kata-kata dari kiri. Mekanisme utama dari algoritma Boyer Moore selain looking-glass technique adalah “the character-jump technique”. Dari namanya saja kita dapat mengartikan bahwa teknik ini dapat melakukan skip/lompat pada beberapa kasus pencocokan string. Algoritma ini menerapkan prinsip akhiran yang baik (*good suffix*), dimana karakter yang dicari sesuai dengan karakter yang dimilikinya, serta prinsip karakter buruk (*bad character*), yang jika karakter tidak memiliki kesamaan akan segera diloncati. Algoritma ini memiliki kompleksitas algoritma $O(nm+A)$ untuk kasus terburuknya. Walaupun kompleksitas algoritma lebih besar jika dibandingkan dengan *bruteforce* maupun KMP, tetapi algoritma Boyer Moore memiliki kelebihan yang lebih besar jika dibandingkan dengan algoritma lainnya. Algoritma ini dapat mencari string atau *pattern* pada suatu teks jauh lebih cepat untuk variasi karakter yang beragam. Keuntungan inilah yang membuat algoritma ini sering dipakai untuk melakukan string matching karena pada kehidupan sehari-hari karakter yang berada pada suatu teks pasti sangat bervariasi. Kelemahan dari algoritma ini adalah ketika variasi karakter yang dicari tidak beragam seperti karakter binary.

2.3 Reguler Expression

Reguler Expression atau biasa disebut regex adalah sebuah teks atau string yang mendefinisikan sebuah pola pencarian sehingga dapat membantu kita untuk melakukan pencocokan, pencarian, dan manipulasi teks.

2.4 Penjelasan singkat mengenai Chatbot

Pada kali ini, chatbot yang dibuat dapat menerima pesan kita dan menjadikan *reminder* dari teks yang kita ketik. Teks terbagi menjadi tanggal, mata kuliah, jenis tugas, dan topik tugas. Chatbot dapat menambahkan *task* baru, menampilkan daftar *task* yang harus dikerjakan, memperbarui *task* tertentu, menandai jika *task* sudah selesai dikerjakan, menampilkan opsi fitur, dan menampilkan pesan *error* jika pesan yang didapat tidak sesuai. Aplikasi ini berjalan dengan kakas JavaScript.

Bab 3

Analisis Pemecahan Masalah

3.1 Langkah penyelesaian masalah setiap fitur

Pada permasalahan menambahkan *task* baru, ketik jenis tugas, mata kuliah, jenis tugas, *keyword* pada sebagai pemisah string dan tanggal, dan tanggal deadline.

Pada permasalahan menampilkan daftar *task* yang harus dikerjakan, setelah memasukkan *task* yang akan dikerjakan, pengguna cukup mengetikkan tanggal yang berurut sesuai tanggalan dengan format *dd/mm/yyyy* atau *dd/mm/yy*, maka chatbot akan mengeluarkan daftar deadline dengan tanggal terdekat yang paling teratas.

Pada permasalahan memperbarui task tertentu, ketik keyword *Deadline task* lalu nomor ID, dan keyword *diundur menjadi* dan tanggal pengunduran

Pada permasalahan menandai jika *task* sudah selesai dikerjakan, ketik keyword *Saya sudah selesai mengerjakan task* lalu dengan nomor ID

Pada permasalahan menampilkan opsi fitur, ketik dengan keyword *Apa yang bisa assistant lakukan*, maka akan mengeluarkan fitur dan daftar kata penting

Pada permasalahan menampilkan pesan *error* jika pesan yang didapat tidak sesuai dengan kata *Maaf, kata yang anda sebut tidak dikenal*

3.2 Fitur fungsional dan arsitektur Chatbot yang dibangun

Bab 4

Implementasi dan Pengujian

4.1 Spesifikasi teknis program

Program ini dibangun dengan HTML untuk menyusun kerangka halaman website, CSS untuk mendeskripsikan presentasi halaman web, dan JavaScript sebagai fungsionalitas dan pengelola masukan pengguna. Setiap task, disimpan dalam sebuah object javascript dengan key berupa id dari task tersebut. Sementara itu, value dari object tersebut juga berupa object yang memiliki data antara lain, tanggal, kode mata kuliah, topik, dan tipe tugas.

Berikut adalah fungsionalitas utama yang dimiliki oleh program.

```
$(document).ready(function(){
    $("#send-btn").on("click", function(){
        $value = $("#data").val();
        $msg = '<div class="user-inbox inbox"><div class="msg-
header"><p>'+ $value +'</p></div></div>';
        $(".form").append($msg);
        $("#data").val('');

        let daftarIdDeadline = [];

        if(getKodeMatkul($value) && getTanggal($value) &&
getTipeTugas($value) && getTopik($value)){
            const currentTaskId = addTask($value);
            showMessage("[TASK BERHASIL DICATAT]");

            let tugas = task[currentTaskId];
            tugas = [tugas.tanggal, tugas.kodeMatkul, tugas.tipeTugas,
tugas.topik].join(" - ");
            tugas = "(ID: "+currentTaskId+" ) " + tugas;
            showMessage(tugas);
        }else if(isTugasDiundur($value)){
            const id = getIdTugas($value);
            if(id!=null){
                task[id].tanggal = getTanggal($value);
                showMessage("Tugas berhasil diperbarui");
            }else{
                showMessage("Tugas gagal diperbarui, pastikan id tugas
tersedia di daftar deadline");
            }
        }

        }else if(isSelesai($value)){
            const id = getIdTugas($value);
            if(id!=null){
                delete task[id];
                showMessage("Tugas berhasil dihapus");
            }else{
                showMessage("Tugas gagal dihapus, pastikan id tugas
tersedia di daftar deadline");
            }
        }
        }else if(isTanyaDeadline($value)){
            if(isTanyaMinggu($value)){
                const banyakMinggu = getBanyakDurasi($value);
```

```

        for(key of Object.keys(task)){
            if(isInNweeksLater(task[key].tanggal, banyakMinggu)){
                let listDeadline = getIdDeadlineByTask($value,
task[key], key);
                for(deadline of listDeadline)
daftarIdDeadline.push(deadline);
            }
        }
        showTugasbyId(daftarIdDeadline);
    }else if(isTanyaHariIni($value)){
        for(key of Object.keys(task)){
            if(isHariIni(task[key].tanggal)){
                let listDeadline = getIdDeadlineByTask($value,
task[key], key);
                for(deadline of listDeadline)
daftarIdDeadline.push(deadline);
            }
        }
        showTugasbyId(daftarIdDeadline);
    }else if(isTanyaHari($value)){
        const banyakHari = getBanyakDurasi($value);
        for(key of Object.keys(task)){
            if(isInNDaysLater(task[key].tanggal, banyakHari)){
                let listDeadline = getIdDeadlineByTask($value,
task[key], key);
                for(deadline of listDeadline)
daftarIdDeadline.push(deadline);
            }
        }
        showTugasbyId(daftarIdDeadline);
    }else if(isAdaRentang($value)){
        const rentangTanggal = getRentangTanggal($value);
        for(key of Object.keys(task)){
            if(isInRentangTanggal(task[key].tanggal,
rentangTanggal)){
                let listDeadline = getIdDeadlineByTask($value,
task[key], key);
                for(deadline of listDeadline)
daftarIdDeadline.push(deadline);
            }
        }
        showTugasbyId(daftarIdDeadline);
    }else if(isTanyaSemuaDeadline($value)){
        for(key of Object.keys(task)){
            daftarIdDeadline.push(key);
        }showTugasbyId(daftarIdDeadline);
    }else if(isAdaKodeMatkul($value)){
        for(key of Object.keys(task)){
            if(task[key].tipeTugas=="Tubes" ||
task[key].tipeTugas=="Tucil"){
                if(task[key].kodeMatkul ==
getKodeMatkul($value)){
                    daftarIdDeadline.push(key);
                }
            }
        }
        showTugasbyId(daftarIdDeadline);
    }
}

```

```

    }else if(isHelp($value)){
        showMessage("[Fitur]");
        showMessage("1. Menambahkan task baru");
        showMessage("2. Melihat daftar task");
        showMessage("3. Melihat deadline task");
        showMessage("4. Memperbarui task");
        showMessage("5. Menghapus finished task");

        showMessage("[Daftar kata penting]");
        showMessage("1. Kuis");
        showMessage("2. Ujian");
        showMessage("3. Tubes");
        showMessage("4. Tucil");
        showMessage("5. Praktikum");

    }else{
        showMessage("Maaf, kata yang anda sebut tidak dikenal");
    }
    });
});

```

Fungsi diatas akan dipanggil ketika sebuah pesan dikirim. Selanjutnya, pesan yang dikirim akan dievaluasi dan diproses sesuai kondisi yang ada. Proses pencarian keyword tertentu dicari menggunakan algoritma boyer-moore, seperti kata “deadline”, “hari”, dan “minggu”. Sedangkan, kata yang memiliki pola tertentu dicari menggunakan bantuan regular expression, seperti tanggal dan kode matakuliah.

4.2 Penjelasan tata cara penggunaan program

Beberapa fitur pada progtram ini adalah:

1. Menambahkan task baru

Untuk menambahkan task baru, pastikan tercantum:

- Kode matakuliah berformat XYYYYY, dengan X adalah huruf dan Y adalah angka.
- Kata penting, yang meliputi kuis, tubes, tucil, ujian, praktikum.
- Topik tugas. Pastikan topik tugas berada setelah kode matakuliah dan sebelum kata "pada".
- Tanggal yang berformat dd/mm/yy. dd/mm/yyyy, tanggal bulan tahun. Contoh format valid: 28 Sept 2022, 28 September 2022, 28/09/22, 28/09/2022.

2. Melihat daftar task

Adapun yang dapat ditampilkan:

- Seluruh task yang sudah tercatat oleh assistant
Contoh perintah yang dapat digunakan: “Apa saja deadline yang dimiliki sejauh ini?”
- 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?”

- 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

3. Melihat deadline task

Hanya berlaku untuk task yang bersifat Tugas atau memiliki tenggat waktu, tugas disini meliputi tubes dan tucil

Misalnya: “Deadline tugas IF2211 itu kapan?”

4. Memperbarui task

Untuk memperbarui deadline, pastikan terdapat kata kunci diundur dan menyebutkan id tugas terkait beserta tanggal baru.

Misal “Deadline task 2 diundur menjadi 28/04/2021”

5. Menghapus finished task Untuk menyelesaikan suatu tugas, pastikan terdapat kata kunci selesai atau menyelesaikan dan menyebutkan id tugas terkait.

Misal “Saya sudah menyelesaikan task 2”

a.

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?”

- 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

6. Melihat deadline task

Hanya berlaku untuk task yang bersifat Tugas atau memiliki tenggat waktu, tugas

disini meliputi tubes dan tucil

Misalnya: "Deadline tugas IF2211 itu kapan?"

7. Memperbarui task

Untuk memperbarui deadline, pastikan terdapat kata kunci diundur dan menyebutkan id tugas terkait beserta tanggal baru.

Misal "Deadline task 2 diundur menjadi 28/04/2021"

8. Menghapus finished task

Untuk menyelesaikan suatu tugas, pastikan terdapat kata kunci selesai atau menyelesaikan dan menyebutkan id tugas terkait.

Misal "Saya sudah menyelesaikan task 2"

4.3 Hasil pengujian (screenshot antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur dalam Chatbot)

1. Menambahkan *task* baru

tubes if2211 string match pada 28/04/2021

[TASK BERHASIL DICATAT]

(ID: 1) 28/04/2021 - IF2211 - Tubes - String Match

2. Menampilkan daftar *task*

apa saja deadline pada tanggal 20/04/2021 hingga pada tanggal 30/04/2021

[Daftar Deadline]

(ID: 1) 28/04/2021 - IF2211 - Tubes - String Match

(ID: 2) 29/04/2021 - IF2200 - Tubes - Coba Tebak

3. Memperbarui task tertentu

deadline task 1 diundur menjadi 28/04/2021

Tugas berhasil diperbarui

kapan deadline if2211?

[Daftar Deadline]

(ID: 1) 28/04/2021 - IF2211 - Tubes - String Match

4. Menandai jika *task* sudah selesai dikerjakan

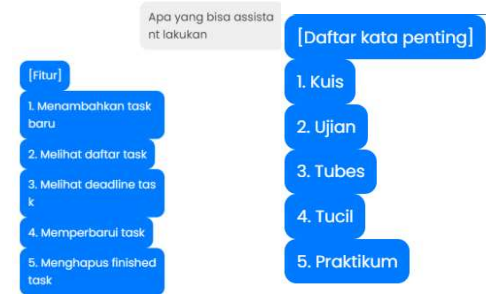
saya sudah menyelesaikan task 1

Tugas berhasil dihapus

kapan deadline if2211

Tidak Ada

5. Menampilkan opsi fitur



6. Menampilkan pesan *error*



4.4 Analisis hasil pengujian

Berdasarkan pengujian yang kami coba, dari keenam parameter, semuanya dapat berhasil dilalui.

Parameter yang dimaksud adalah menambah task baru,

Bab 5

Kesimpulan

5.1 Kesimpulan

Dengan Algoritma Knuth-Morris-Pratt, Algoritma Boyer Moose, dan Regular Expression dapat membuat chatbot yang dapat diajak bercapak, menjawab pertanyaanm dan memanfaatkan chatbot dengan salah satunya dijadikan untuk mengingatkan deadline kita, dapat bertanya kapan sautu deadline, dan mengurutkan jadwal deadline dengan berdasarkan tanggal terdekat.

5.2 Saran

Mengaplikasikan ilmu yang kami punya untuk aplikasi yang dapat kami gunakan sehari-hari sangatlah seru dan meanntang. Semoga ada tugas-tugas lainnya yang memanfaatkan materi yang telah dipelajari agar tidak sekedar menjadi hafalan.

5.3 Komentor

Kami senang dapat mengaplikasikan pembelajaran kedalam aplkasi yang dapat dipakai sehari-hari. Misalnya, mengotak-atik RegEx membuat kami merasa senang dan tentu saja terdapat tantangan tersendiri di dalamnya. Sebab itu, terdapat kepuasan tersendiri ketika berhasil menyelesaikan tugas ini.

Daftar Pustaka

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