Penerapan Algoritma Pencocokan String dan *Regular Expression* dalam Penyensoran Kata pada Permainan Mobile Legends Bang Bang

Raffi Fadhlurrahman Putra Rahiem 13519219
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519219@std.stei.itb.ac.id

***Abstract*—Permainan daring terbagi menjadi dua jenis berdasarkan jumlah pemain yang dibutuhkan yaitu individu dan berkelompok. Mobile Legends: Bang Bang merupakan salah satu permainan daring yang membutuhkan kelompok pemain yang terbentuk dari tiap pemain solo ataupun beregu untuk dapat memainkannya sehingga secara tidak langsung akan dibutuhkan kerjasama dan kekompakan antarindividu serta komunikasi baik dari percakapan teks dalam gim maupun suara dari microphone yang terkadang dapat mengandung umpatan dan sebagainya. Makalah ini dibuat untuk menjelaskan salah satu penerapan algoritma pencocokan string dan regex yaitu penyensoran kata dalam kalimat percakapan pada permainan Mobile Legends: Bang Bang dengan membuat sebuah simulasi sederhana. Terdapat beberapa kasus yang diuji dalam makalah ini untuk menggambarkan berbagai macam contoh situasi yang biasa terjadi di dalam permainan. Hasil dari pengujian simulasi ini menunjukkan bahwa beberapa string yang diharapkan berhasil disensor.**

Keywords—Regex, Pola, Teks, Pencocokan String, Chat, String.

I. PENDAHULUAN

Perkembangan gim ponsel bergenre MOBA (Multiplayer Online Battle Arena) sangatlah pesat, salah satu aplikasi gim yang kini sedang ramai dimainkan kawasan Asia Tenggara, terutama Indonesia ialah Mobile Legends : Bang Bang atau yang biasa disingkat dengan MLBB.

Sumber gambar : Dokumen Pribadi

Karena antusias para pemain yang tinggi, tim pengembang gim ini mulai menambahkan mode-mode arkade baru selain mode biasa seperti Survival, Mayhem, Mirror, Chess Tower Defense (Chess TD), Frenzy, Magic Chess, dan mode lainnya yang selalu hadir dalam waktu terbatas pada event-event besar dan khusus seperti perayaan hari besar, ulang tahun dan waktu lainnya.

Pada tahun awal perilisannya, fitur percakapan antar pemain dalam gim ini belum tersedia sehingga suasana dan atmosfer permainan masih bersifat perseorangan dan kurang adanya kerjasama sebagai imbas dari tidak tersedianya wadah komunikasi. Kemudian pada perilisan global, fitur chat ditambahkan dalam permainan sehingga para pemain bisa saling bertukar pikiran dan informasi saat permainan berlangsung.

Sumber gambar : https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.reddit.com%2Fr%2Fmobilelegends%2Fcomments%2Fghczcf%2Flook_the_reason_for_the_mute_mobile_legends_keeps%2F&psig=AOvVaw1clZP8R_HIh-HnfsZs638m&ust=1620838264461000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPjA85qLwvACFQAAAAAdAAAAABAD

Seiring dengan bertambahnya pemain yang mendaftar dan bermain dalam permainan dengan sifat dan kepribadian yang berbeda, suasana permainan menjadi beragam. Ada kalanya percakapan antar pemain dalam permainan berisi tentang hujatan dan kata kasar, pujian atas performa pemain lain, dan topik lainnya sehingga dibuatlah batasan tertentu untuk menjaga lingkungan permainan yang bersih dan sportif. Salah

satu langkah yang diambil oleh pihak pengembang adalah dengan melakukan penyensoran kata yang dianggap kasar dan pemberian sanksi atau hukuman terhadap pemain yang telah melewati batas yang ditentukan dalam berkomunikasi dengan pemain lain.

II. Landasan Teori

A. *Pencocokan String*

Pencocokan string atau algoritme pencarian string adalah algoritme untuk melakukan pencarian semua kemunculan string pendek yang disebut pattern di string yang lebih panjang yang disebut teks.

Pencocokan String telah digunakan pada berbagai macam kegiatan seperti :

1. Pencarian di dalam editor teks

Salah satu fitur yang biasa ditemukan pada sebuah text editor adalah fitur pencarian kata. Fitur ini menggunakan algoritma pencocokan string untuk menemukan string kata dari masukan pengguna di dalam suatu file.

2. Mesin pencari situs

Mesin pencari situs seperti Google, Bing, dan sebagainya menggunakan algoritma pencocokan string untuk menampilkan hasil pencarian situs terurut dari nilai kecocokan terbesar.

3. Analisis Citra

Analisis Citra menggunakan konsep pencocokan string dalam menentukan membandingkan wujud penampakan dua objek.

4. Bioinformatika

Bioinformatika adalah ilmu yang mempelajari penerapan teknik komputasional untuk mengelola dan menganalisis informasi biologis. Bidang ini mencakup penerapan metode-metode matematika, statistika, dan informatika untuk memecahkan masalah-masalah biologis, terutama dengan menggunakan sekuens DNA dan asam amino serta informasi yang berkaitan dengannya. Contoh topik utama bidang ini meliputi basis data untuk mengelola informasi biologis, penyejajaran sekuens (sequence alignment), prediksi struktur untuk meramalkan bentuk struktur protein maupun struktur sekunder RNA, analisis filogenetik, dan analisis ekspresi gen.

String merupakan sebuah rangkaian yang terdiri dari karakter tertentu. Dalam hal ini, jika diasumsikan S sebuah string dengan ukuran m dan k indeks sembarang antara 0 dan m-1,

$S = x_0x_1\ldots x_m$

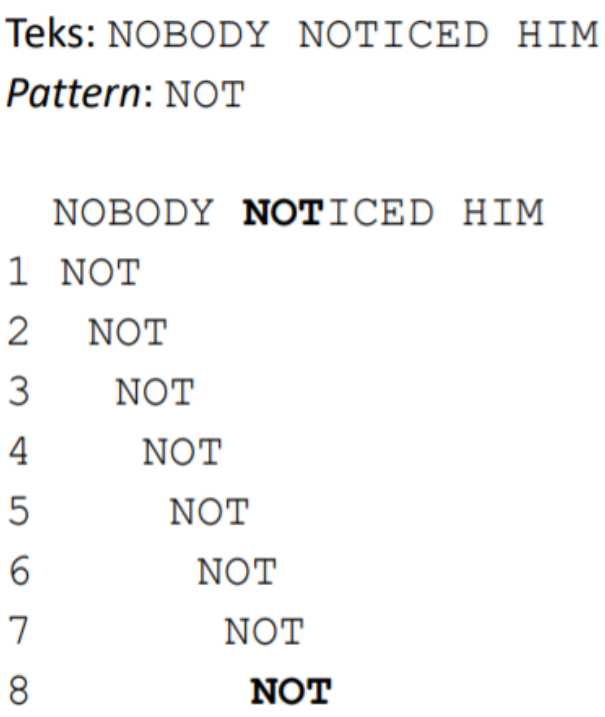
Maka prefix dari S merupakan bagian rangkaian S[0..k] dan suffix dari S merupakan bagian rangkaian S[k..m-1]. Misal S = mata, maka semua prefix yang mungkin dari S adalah “m”, “ma”, “mat”, dan “mata” sedangkan semua suffix yang mungkin dari S adalah “a”, “ta”, “ata”, dan “mata”.

Pencocokan String memiliki banyak macam jenis algoritma yaitu :

1. Algoritma Brute Force

Algoritma Brute Force adalah pendekatan yang lempang (straightforward) untuk memecahkan suatu persoalan yang biasanya didasarkan pada pernyataan pada persoalan dan definisi/konsep yang terlibat sehingga memecahkan masalah dengan cara yang sangat sederhana, langsung, dan jelas.

Dalam kasus pencocokan string, algoritma ini berarti mengecek tiap posisi dalam teks T untuk melihat jika pola P mulai dari posisi tersebut.

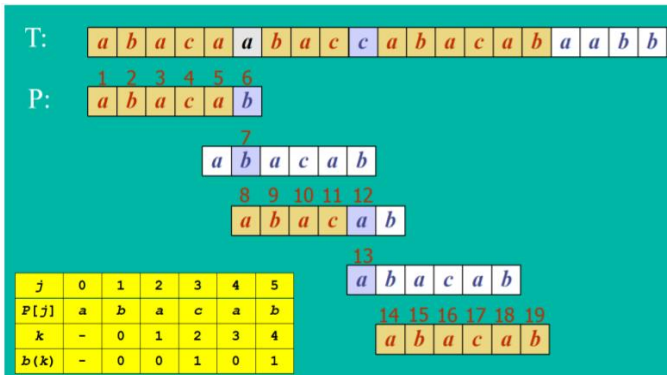


Sumber gambar :
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

Gambar diatas menggambarkan bahwa pencocokan dilakukan dengan menguji kesamaan pada perbandingan 1, jika tidak cocok maka pola digeser sejauh satu karakter untuk dibandingkan kembali dan berulang hingga ditemukan kecocokan pola terhadap teks.

2. Algoritma KMP

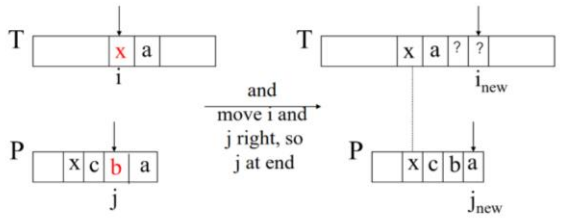
Algoritma Knuth–Morris–Pratt, disingkat KMP, merupakan algoritma pencocokan string yang mencari lokasi pola string dalam teks. Algoritma ini mencari dari kiri ke kanan hingga menemukan kecocokan yang tepat seperti brute force, namun memiliki perbedaan dalam teknik pergeseran.



Jika terdapat ketidakcocokan pada posisi ke-j dari pola P, maka pencarian pada karakter baru dimulai pada indeks prefix terbesar dari pola P hingga posisi ke-(j-1) yang merupakan suffix dari pola P hingga posisi ke-(j-1), tidak termasuk karakter pertama. Dalam kata lain, jika P[j] ≠ Teks[i], algoritma membandingkan P dan Teks pada posisi j' dimana P[0...j'] = P[j-1-j'...j-1], dimana j' ≤ j-1.

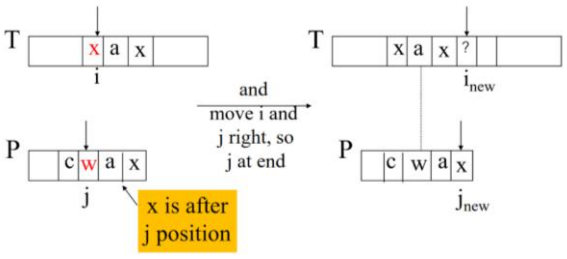
3. Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan algoritma pencarian string dalam teks dengan melakukan pencocokan string dari sebelah kanan pola. Misal terjadi ketidakcocokan pada posisi ke-i pada teks yaitu karakter "x" yang tidak sama dengan karakter ke-j pada pola P.



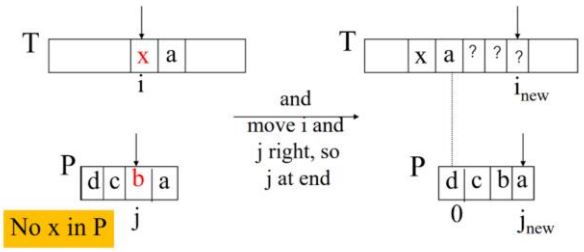
Sumber gambar :
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

Jika pola P mengandung karakter "x" namun posisinya berada di kiri posisi ke-j maka pola P digeser sedemikian rupa sehingga posisi karakter "x" pada pola P sejajar dengan posisi karakter "x" pada teks dan posisi ke-i baru pada teks dipindah menjadi sejajar dengan posisi paling kanan pola P yang menjadi posisi ke-j baru.



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

Jika pola P mengandung karakter "x" namun posisinya berada di kanan posisi ke-j maka pola P digeser satu karakter ke kanan dan posisi ke-i baru pada teks dipindah menjadi sejajar dengan posisi paling kanan pola P yang menjadi posisi ke-j baru.



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

Jika pola P tidak mengandung karakter "x" maka pola P digeser sedemikian rupa sehingga posisi pola P sejajar dengan posisi ke-i+1 pada teks dan posisi ke-i pada teks dipindah menjadi sejajar dengan posisi paling kanan pola P yang menjadi posisi ke-j baru.

4. Regular Expression

B. Regular Expression

Regular Expression atau yang disingkat sebagai regex atau regexp adalah sebuah rangkaian yang terdiri dari karakter yang menentukan sebuah pola pencarian. Regex biasa digunakan pada algoritma pencarian string dalam menemukan dan mengganti string serta dalam digunakan dalam validasi suatu masukan.

Setiap karakter dalam regex memiliki arti khusus, beberapa diantaranya ialah :

Metakarakter	Makna
.	Semua karakter kecuali *newline*
\.	Tanda baca titik (Berlaku pula untuk tanda baca lain, contohnya *, \{, \\, dll.)
^	Awal string
$	Akhir string
\d, \w, \s	Angka, angka & huruf [A-Za-z0-9_], spasi
\D, \W, \S	Semua karakter kecuali angka, angka & huruf [A-Za-z0-9_], spasi
[abc]	Karakter a, b, atau c
[a-z], [A-Z]	Karakter dari a-z (huruf kecil) , karakter dari A-Z (huruf kapital)

[^abc]	Semua karakter kecuali a, b, atau c
aa | bb	Karakter dengan pola aa atau bb
?	Karakter sebelumnya berjumlah nol atau satu
*	Karakter sebelumnya berjumlah nol atau lebih
+	Karakter sebelumnya berjumlah satu atau lebih
{*n*}	Karakter sebelumnya berjumlah *n*
{*n*,}	Karakter sebelumnya berjumlah *n* atau lebih
{*m*, *n*}	Karakter sebelumnya berjumlah antara *m* dan *n*
??, *?, +?, {*n*}?, dll	Sama seperti sebelumnya, tetapi jika karakter telah ditemukan, pencarian berhenti
(*expr*)	Melakukan *grouping* terhadap ekspresi
(?:*expr*)	Melakukan *grouping* juga, tetapi tidak di-*capture*
(?=*expr*)	Diikuti oleh ekspresi
(?!*expr*)	Tidak diikuti oleh ekspresi

C. *In-game Chat*

Menurut Kamus Oxford, Chat atau obrolan adalah percakapan yang dilakukan dengan cara informal. Dalam gim, obrolan biasanya berbasis teks atau suara. Pemain dapat mengetik pesan atau mengirimkan suara melalui game dan orang lain dalam game dapat memilih untuk membaca atau mendengarnya. Obrolan dalam game tidak melibatkan video sehingga antarpemain tidak akan dapat melihat wajah lawan bicara saat mengobrol.

Obrolan dalam gim memungkinkan pemain mengatakan apa saja, dengan beberapa batasan untuk membuat pengalaman gim lebih menyenangkan bagi semua pemain.

Obrolan teks dalam game di sebagian besar game memiliki semacam filter sumpah serapah di dalamnya, sehingga pemain tidak dapat merusak suasana dengan umpatan yang berlebihan. Obrolan teks dalam game mungkin juga memiliki pembatas 'spam'. Pemain yang memposting terlalu sering dapat diblokir sementara dari menggunakan obrolan dalam game selama 20 hingga 30 detik. Pemain yang terus-menerus memasukkan kata-kata kosong ke keyboard kemungkinan besar akan menerima blok spam sementara. Pemain masih dapat memainkan game tersebut saat mereka diblokir dari obrolan teks. Obrolan suara tidak terlalu dibatasi sehingga pemain dapat mengatakan apa saja melalui microphone.

Sumber gambar : Dokumen Pribadi

Selain itu, beberapa gim menyediakan fitur obralan cepat berupa kumpulan tanggapan umum yang biasa diucapkan pada antarpemain sehingga mempercepat penanggapan dalam permainan dan pemain tidak perlu repot mengetik.

III. PEMBAHASAN

A. *Simulasi In-Game Chat*

Pada program ini, penulis menggunakan bahasa pemrograman Python. Daftar Regular Expression yang digunakan pada simulasi kali ini adalah sebagai berikut :

1. [fF][\w]*[uUcC]*[\w]*[kK]

Contoh string yang cocok dengan regex ini adalah “fucek”, “fck”, dan sebagainya.

2. [aA4][sS5]{2}

Contoh string yang cocok dengan regex ini adalah “A55”, “4SS”, dan sebagainya.

3. [bB][a-zA-Z4]*[gG][\w]*[sS5][a-zA-Z4]*[tT]

Contoh string yang cocok dengan regex ini adalah “bgst”, “bgsat”, dan sebagainya.

4. [nN][oOuU0]*[bB]

Contoh string yang cocok dengan regex ini adalah “NoOB”, “nub”, “nb”, dan sebagainya.

5. [jJ][a-zA-Z4]*[nN][a-zA-Z]*[cC][a-zA-Z0]*[kK]

Contoh string yang cocok dengan regex ini adalah “juancik”, “jnck”, dan sebagainya.

6. [kK][a-zA-Z40]*[nN][a-zA-Z4]*[tT][a-zA-Z41]*[Ll]

Contoh string yang cocok dengan regex ini adalah “kintal”, “kental”, dan sebagainya.

7. [aA][a-zA-Z]*[nN][jJ][a-zA-Z]*[gG]")]

Contoh string yang cocok dengan regex ini adalah “anjg”, “anjing”, dan sebagainya.

Secara garis besar, simulasi ini akan melakukan beberapa hal yaitu :

1. Menginisialisasi daftar pola yang digunakan
2. Meminta input teks dari pengguna
3. Melakukan proses pencarian untuk menemukan string yang cocok dengan tiap pola yang digunakan.
4. Menyimpan tiap string yang cocok dengan pola pada list A.
5. Mencari lokasi string dengan melakukan proses pencocokan dengan list A kemudian menggantinya

dengan karakter “*” sebanyak panjang string yang disimpan pada list A.

6. Menampilkan kembali teks yang telah disensor ke layer.

Kode program yang digunakan pada simulasi ini ialah sebagai berikut :

```
import re

valid = 1
mute = 0
count = 0
chat = [("
A : Sabar \n"),
        ("
A : Udah jangan marah, main santai aja
\n"),
        ("
A : Berisik \n")]

print ("~~~~~~~~~~~~~~~~~~~~YOU HAVE
BEEN SLAIN~~~~~~~~~~~~~~~~~~~~\n")
while (valid == 1) :
    txt = input(chat[count%2])
    pattern =
[("[fF][\w]*[uUcC]*[\w]*[kK]"),
("[aA4][sS5]{2}"), ("[bB][a-zA-
Z4]*[gG][a-zA-Z]*[sS5][a-zA-Z4]*[tT]"),
                ("[nN][oOuU0]*[bB]"),
("[jJ][a-zA-Z4]*[nN][a-zA-Z]*[cC][a-zA-
Z0]*[kK]"), ("[kK][a-zA-Z40]*[nN][a-zA-
Z4]*[tT][a-zA-Z41]*[Ll]"),
                ("[aA][a-zA-
Z]*[nN][jJ][a-zA-Z]*[gG]")]
    final = txt
    bad = []
    for i in range (len(pattern)) :

bad.append(re.findall(pattern[i], txt))
    for i in range (len(bad)):
        for j in range (len(bad[i])):
            final = re.sub(bad[i][j],
"*"*len(bad[i][j]), final)
            mute += 1
    if (mute < 5):
        print ("
B : " +final)
        count += 1
    else :
        print ("
B : " +final)
        print ("
[System]Serious Inappropriate Chatting
behavior has been detected. \n
You have been muted! (Please check in-
game mail)")
        valid = 0
```

B. Pengujian berbagai kasus

a. Kasus 1 :

```
~~~~~~~~~~~~~~~~~~~~YOU HAVE BEEN SLAIN~~~~~~~~~~~~~~~~~~~~
```

```
A : Sabar
```

Balasan 1: fuck you

```
B : **** you
A : Udah jangan marah, main santai aja
```

Analisis :

Ketika program berjalan, program akan meminta input balasan. Pada kasus 1, teks balasan berisi “fuck you”. Kemudian program akan melakukan pencarian string teks dengan tiap pola yang telah diprogram dan jika ditemukan string dengan pola yang sesuai maka string akan disimpan pada list bad. Pada kasus 1, ditemukan string yang sesuai dengan pola pada teks yaitu “fuck”. Selanjutnya, program akan melakukan penyensoran string yang melakukan pencarian stirng kembali dengan pattern yang sesuai pada list bad dan mengganti dengan karakter “*” sebanyak panjang tiap string pada list bad. Pada kasus 1, string “fuck” dengan panjang 4 karakter diganti menjadi string dengan 4 karakter “*” sehingga respon B yang muncul pada simulasi adalah “**** you”.

b. Kasus 2 :

```
~~~~~~~~~~~~~~~~~~~~YOU HAVE BEEN SLAIN~~~~~~~~~~~~~~~~~~~~
```

```
A : Sabar
```

Balasan 1 : mau sabar gimana anjing dapat team kek kintil semua

```
B : mau sabar gimana ****** dapat team kek ****** semua
A : Udah jangan marah, main santai aja
```

Balasan : emang dasar moonton juancik bingset dpt team nub

```
B : emang dasar moonton ****** ******* dpt team ***
[System]Serious Inappropriate Chatting behavior has been detected.
You have been muted! (Please check in-game mail)
```

Analisis :

Ketika program berjalan, program akan meminta input balasan 1. Pada kasus 2, teks balasan 1 berisi “mau sabar gimana anjing dapat team kek kintil semua”. Kemudian program akan melakukan pencarian string teks dengan tiap pola yang telah diprogram dan jika ditemukan string dengan pola yang sesuai maka string akan disimpan pada list bad. Pada kasus 2, ditemukan string yang sesuai dengan pola pada teks balasan 1 yaitu “anjing” dan “kintil”. Selanjutnya, program akan melakukan penyensoran string yang melakukan pencarian stirng kembali dengan pattern yang sesuai pada list bad dan mengganti dengan karakter “*” sebanyak panjang tiap string pada list bad. Pada kasus 2 balasan 1, string “anjing” dengan panjang 6 karakter diganti menjadi string dengan 6 karakter “*” dan string “kintil” dengan panjang 6 karakter diganti menjadi string dengan 6 karakter “*” sehingga respon yang muncul pada simulasi adalah “mau sabar gimana ****** dapat team kek ****** semua.

Ketika program berlanjut dengan input balasan 2. Pada kasus 2, teks balasan 2 berisi “emang dasar moonton juancik bingset dpt team nub”. Kemudian program akan melakukan pencarian string teks dengan tiap pola yang telah diprogram dan jika ditemukan string dengan pola yang sesuai maka string akan disimpan pada list bad. Pada kasus 2, ditemukan string yang sesuai dengan pola pada teks balasan 2 yaitu “juancik”, “bingsit”, dan “nub”. Selanjutnya, program akan melakukan penyensoran string yang melakukan pencarian stirng kembali dengan pattern yang sesuai pada list bad dan mengganti dengan karakter “*” sebanyak panjang tiap string pada list bad. Pada kasus 2 balasan 2, string “juancik” dengan panjang 7 karakter diganti menjadi string dengan 7

karakter “*” dan string “bingsit” dengan panjang 7 karakter diganti menjadi string dengan 7 karakter “*” serta string “nub” dengan panjang 3 karakter diganti menjadi string dengan 3 karakter “*” sehingga respon yang muncul adalah “emang dasar moonton ******* ******* dpt team ***.

Kemudian karena program menemukan bahwa pemain B telah melakukan pelanggaran lebih dari batas yang telah diprogram maka program menampilkan pesan peringatan dan mengakhiri program simulasi sebagai bentuk hukuman mute pada pemain B.

IV. KESIMPULAN

Dari pembahasan dapat disimpulkan bahwa ilmu Strategi Algoritma, terutama topik Pencocokan String dengan Regular Expression dapat diterapkan dalam melakukan penyensoran kata yang kasar, umpatan tertentu, dan sebagainya yang bermakna kurang baik pada teks masukan yang telah diuji. Selain ini, konsep algoritma pencocokan string juga telah digunakan dalam berbagai ilmu pengetahuan yaitu dalam ilmu biologi seperti bioinformatika, sosial seperti penyensoran kata yang kurang pantas, astronomi, sejarah, ekonomi, dan sebagainya

V. PENUTUP

Konsep berpikir tentang menentukan kebenaran sebuah berita juga secara tidak langsung telah menerapkan konsep dari Algoritma Pencocokan String. Penulis berharap agar para pembaca dapat lebih terbuka pemikirannya dan tidak malas dalam berpikir dengan konsep Algoritma Pencocokan String dan menerapkannya dalam kehidupan sehari-hari.

UCAPAN TERIMA KASIH

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada Allah SWT. karena atas izinnya penulis dapat menyelesaikan tugas makalah ini. Ucapan terima kasih juga ingin disampaikan oleh penulis kepada orang tua penulis atas dukungannya selama ini, serta kepada seluruh dosen mata kuliah Strategi Algoritma, terutama kepada Bapak Rinaldi Munir selaku dosen pengajar kelas K04, atas ilmu yang telah disampaikan.

DAFTAR REFERENSI

[1] Lecroq, Thierry Charras, Christian. 2001. Handbook of Exact String Matching Algorithm. ISBN 0-9543006-4-5

[2] Tim pengajar Program Studi Teknik Informatika STEI-ITB. Pencocokan String (String/Pattern Matching) https://informatika.stei.itb.ac.id /~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf, diakses pada 11 Mei 2021

[3] Khodra, Masayu Leylia. String Matching dengan Regular Expression ,https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf, diakses pada 11 Mei 2021

[4] Tim pengajar Program Studi Teknik Informatika STEI-ITB. Algoritma Brute Force)https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Brute-Force-(2021)-Bag1.pdf, diakses pada 11 Mei 2021

[5] NSW Department of Education, “What do I need to know about in-game chat?”,https://www.digitalcitizenship.nsw.edu.au/articles/what-do-i-need-to-know-about-in-game-chat#:~:text=In%2Dgame%20chat%20is%20usually,people%20you%20are%20chatting%20with, diakses pada 11 Mei 2021

[6] Resa Risyan, 2019, “Apa Itu Text Editor, Fungsi Dan Contohnya (LENGKAP)”,https://www.monitorteknologi.com/apa-itu-text-editor/#2_Mencari_Kata_Pada_Sebuah_File_Ataupun_Folder, diakses pada 11 Mei 2021

[7] Susilawati dan Bachtiar, N. (2018). Biologi Dasar Terintegrasi (PDF). Pekanbaru: Kreasi Edukasi. hlm. 4. ISBN 978-602-6879-99-8

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Mei 2021

Raffi Fadhlurrahman Putra Rahiem 13519219