Penerapan Algoritma Pencocokan String dan Regex dalam Mendeteksi Konten Berbahaya pada Media Sosial

Jonathan Emmanuel Saragih - 13522121
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): jonathan.srgh@gmail.com

***Abstract*—Dalam masa yang semakin modern dan perkembangan jaman yang semakin pesat, media sosial menjadi salah satu hal yang dimiliki oleh semua orang. Seluruh golongan Masyarakat mulai dari yang berstatus sosial tinggi maupun rendah, berada pada kota besar atau kota kecil, usia tua maupun muda, semua memiliki media sosial. Dengan adanya diversitas ini, serangan pengaruh buruk serta konten yang berbahaya menjadi salah satu permasalahan yang kita hadapi bersama. Dengan adanya tantangan ini, muncul kejahatan cyber seperti perundungan, ujranan kebencian, disinformasi, dan masih banyak kejahatan pada dunia maya lainnya. Dalam makalah ini, akan dieksplorasi mengenai penerapan algoritma pencocokan string dan regex dalam mendeteksi konten berbahaya secara efektif. Pada makalah ini, akan diteliti lebih jauh mengenai algoritma Knuth Morris Pratt dan Boyer Moore dalam mengecek dataset atau kasus yang memiliki data besar pada media sosial. Selain itu, dengan adanya regex, pendeteksian akan menjadi semakin fleksibel dan efisien.**

Keywords—Algoritma Pencocokan String, Ekspresi Reguler (Regex), Deteksi Konten Berbahaya, Ujaran Kebencian, Perundungan Siber, Knuth-Morris-Pratt (KMP), Boyer-Moore(BM), Media Sosial, Keamanan online

I. PENGENALAN

Dalam dunia yang semakin modern dan maju, media sosial menjadi bagian yang tidak terpisahkan dari tiap tiap individu. Media sosial menjadi sangat erat kaitannya dengan gaya hidup masyarakat saat ini. Media sosial sendiri merupakan plaform yang menyediakan ruang bagi individu untuk berkomunikasi, saling berbagi informasi, berbagi cerita, mengekspresikan diri, dan masih banyak hal lainnya. Namun dengan luasnya kebebasan yang dimiliki oleh para pengguna media sosial, hal ini memicu tantangannya sendiri. Konten berbahaya, seperti ujaran kebencian, perundungan siber, dan disinformasi, dapat merusak keharmonisan sosial, mengganggu kesejahteraan individu, dan bahkan mengancam keselamatan publik.

Untuk mengatasi masalah yang berat ini, perlu ada mekanisme atau fasilitas keamanan yang efektif dan efisien untuk dapat mendeteksi dan mengendalikan penyebarluasan konten berbahaya yang ada pada media sosial/ Salah satu cara atau penggunakan algoritma yang efektif adalah algoritma pencocokan string dan regex. Kedua algoritma ini memiliki tingkat efisiensi yang tinggi dan cepat sehingga dapat mendeteksi pola berbahaya dalam data yang besar, mengingat pengguna media sosial di Indonesia memiliki jumlah yang sangat besar.

Makalah ini ditulis untuk menjadi sarana pembelajaran untuk memperdalam dan mengeksplorasi penerapan algoritma pencocokan string dan regex dalam mendeteksi konten berbahaya di media sosial. Kami akan membahas bagaimana kedua teknik ini dapat digunakan secara efektif untuk mengidentifikasi dan menyaring konten yang melanggar kebijakan komunitas dan mengancam keamanan pengguna. Selain itu, kami juga akan mengkaji efisiensi dan akurasi dari berbagai algoritma pencocokan string, seperti Knuth-Morris-Pratt (KMP) dan Boyer-Moore, dalam konteks media sosial yang dinamis.

Melalui penelitian ini, diharapkan akan dapat memberikan kontribusi yang lebih besar dan memiliki hasil yang cukup signifikan dalam menciptakan lingkungan media sosial yang lebih aman dan nyaman bagi semua pengguna dari segala kalangan.

II. DASAR TEORI

Dalam mendeteksi konten berbahaya untuk media sosial, diperlukan beberapa teori dan algoritma untuk digunakan. Dalam hal ini, diperlukan pemahaman yang lebih mendalam mengenai teknik dalam pemrograman dan komputasi untuk membuatnya. Dalam hal ini, diperlukan pengetahuan mengenai algoritma pencocokan string dan regex. Pada bab 2 yang berisi dasar teori, akan ditelaah lebih dalam mengenai teori yang akan kita gunakan.

A. Algoritma Pencocokan String

Algoritma string memegang peran yang penting dalam hal mencari konten berbahayaa yang ada pada media sosial. Algoritma pencocokan string memiliki tujuan untuk mencari dan menemukan sebuah konten atau pola dalam teks yang berada pada teks lebih panjang. Pada algoritma pencocokan string, akan dibahas mengenai dua algoritma yang lebih spesifik yaitu algoritma Knuth-Morris-Pratt (KMP) dan juga

algoritma Boyer-Moore (B). Kedua algoritma tersebut merupakan algoritma yang sering dan biasa digunakan dalam pencocokan string karena memiliki kelebihannya masing masing.

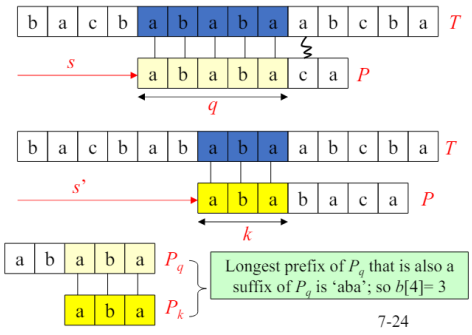
B. Algoritma Knuth-Morris-Pratt (KMP)

Algoritma KMP merupakan algoritma yang dikembangkan oleh Donald Knuth, Vaughan Pratt, dan James H. Morris yang ditemukan pada tahun 1977. Algoritma KMP merupakan algoritma yang dirancang untuk mengatasi ketidakefisienan yang terdapat pada algoritma pencocokan string sederhana atau yang biasa kita kenal sebagai algoritma bruteforce. Algoritma sederhana (bruteforce) akan melakukan perbandingan karakter demi karakter dan akan memulai kembali perbandingan dari awal teks jika ditemukan ketidakcocokan. Proses ini bisa sangat menyita banyak waktu dan tidak efisien jika pola yang dicari memiliki banyak karakter yang mirip dengan teks.

Algoritma KMP memanfaatkan struktur dari pola yang sedang dicari untuk mempercepat proses pencocokan melalui penggunaan "failure function" atau "partial match table." Tabel ini menyimpan informasi tentang panjang prefiks dari pola yang juga merupakan sufiks. Dengan menggunakan tabel ini, KMP dapat menghindari perbandingan ulang yang tidak perlu.

Terdapat 2 langkah pada algoritma KMP yaitu :

1. Preprocessing: Membuat failure function untuk pola. Hal ini dilakukan dengan menghitung panjang prefiks terpanjang yang juga merupakan sufiks untuk setiap posisi dalam pola.
2. Pencocokan: Menggunakan failure function untuk melompati bagian teks yang tidak perlu dibandingkan ulang. Jika ditemukan ketidakcocokan, algoritma menggunakan informasi dari failure function untuk menentukan posisi selanjutnya di mana pencocokan harus dilanjutkan.



Gambar 1. Visualisasi cara kerja algoritma KMP

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

C. Algoritma Boyer-Moore (BM)

Algoritma Boyer-Moore atau yang biasa disingkat dengan algoritma BM merupakan algoritma yang dikembangkan oleh Robert S. Boyer dan J Strother Moore pada tahun 1977. Algoritma Boyer-Moore adalah salah satu algoritma pencocokan string yang paling efisien dalam penggunaannya pada kasus kasus sehari hari. Keunggulan utama dari algoritma ini terletak pada dua heuristik utama yang digunakan yaitu "bad character rule" dan "good suffix rule."

Bad Character Rule: Pada Bad Character Rule, akan berlaku aturan bahwa jika terjadi ketidakcocokan antara karakter dalam pola dan teks, program akan mencari karakter yang tidak cocok dalam pola. Kemudian program akan melompati sejumlah karakter yang sama dengan jarak dari posisi karakter yang tidak cocok dalam pola ke tepi paling kanan dari pola yang cocok. Berikut merupakan rumus yang digunakan :

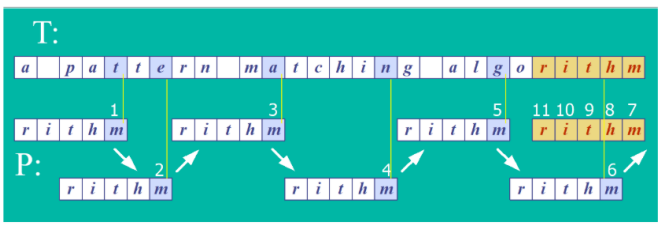
$$Bc[c] = \max\{1, j - last(c)\}$$

Good Suffix Rule: Sebaliknya, pada Good Suffix Rule, berlaku aturan yaitu jika terjadi ketidakcocokan setelah sebagian besar pola telah cocok, program dirancang untuk mencari posisi terakhir dari bagian pola yang cocok dalam teks. Kemudian, program ini akan melompati sejumlah karakter yang sesuai dengan jarak dari posisi terakhir bagian yang cocok ke awal pola yang cocok. Berikut merupakan rumus untuk menentukan Good Suffix Rule :

$$Gs[i] = \min\{shift(j) \mid 0 \le j < m \text{ , } P[j] =/= P[i]\}$$

Terdapat 2 langkah pada algoritma BM yaitu **:**

1. Preprocessing: Membuat tabel untuk bad character rule dan good suffix rule. Tabel bad character akan mencatat posisi terakhir dari setiap karakter dalam pola, sedangkan tabel good suffix mencatat jarak lompatan berdasarkan sufiks yang cocok.
2. Pencocokan: Memulai pencocokan dari akhir pola. Jika terjadi ketidakcocokan, algororitma BM akan menggunakan salah satu dari dua heuristik untuk menentukan lompatan yang optimal.



Gambar 2. Visualisasi cara kerja algoritma BM
Sumber :
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

D. Ekspresi Reguler (Regex)

Ekspresi Reguler atau yang biasa kita kenal dengan Regex adalah sekumpulan karakter yang mendefinisikan pola pencarian tertentu. Regex memiliki kelebihan yaitu sangat kuat dalam mengenali pola teks yang kompleks dan beragam,. Hal ini menjadikan Regex sebagai *tools* yang ideal untuk mendeteksi berbagai jenis konten berbahaya di media sosial. Regex dapat digunakan untuk mencari kata kunci spesifik,

pola kalimat, atau struktur teks tertentu yang sering ditemukan dalam konten berbahaya.

Terdapat beberapa hal yang perlu dipahami leih dalam mengenai Regex , diantaranya :

1. Sintaks Dasar Regex: Regex menggunakan berbagai metakarakter untuk mendefinisikan pola pencarian, seperti titik (.), tanda bintang (*), tanda tanya (?), dan kurung kurawal ({}). Kombinasi metakarakter ini memungkinkan pencarian pola yang sangat spesifik dan fleksibel.
2. Kombinasi dengan Algoritma Pencocokan String: Dalam banyak kasus, regex dapat dikombinasikan dengan algoritma pencocokan string untuk meningkatkan efisiensi dan akurasi. Misalnya, regex dapat digunakan untuk mendefinisikan pola awal yang harus dicari, sementara algoritma pencocokan string dapat digunakan untuk melakukan pencarian sebenarnya dalam teks yang besar.Kaitan antara Teori Pencocokan String dalam Mendeteksi Konten Berbahaya

Penggunaan algoritma pencocokan string dan ekspresi reguler (regex) sangat penting dalam mendeteksi konten berbahaya di media sosial. Konten seperti ujaran kebencian, perundungan siber, dan disinformasi sering kali memiliki pola teks tertentu yang bisa dikenali oleh kedua teknik ini. Dengan mendefinisikan pola-pola ini secara tepat, sistem moderasi konten bisa secara otomatis memindai dan menyaring konten yang melanggar kebijakan komunitas.

Pendekatan ini idak hanya membantu mendeteksi konten berbahaya dengan lebih cepat dan akurat, tetapi juga mengurangi beban kerja moderator manusia. Kombinasi algoritma pencocokan string dan regex memungkinkan penciptaan sistem yang dapat beradaptasi dengan perubahan pola teks yang digunakan dalam konten berbahaya, sehingga tetap efektif dalam jangka panjang.

Selain itu, metode ini bisa mengidentifikasi berbagai bentuk konten berbahaya dengan tingkat akurasi yang tinggi, memastikan bahwa konten yang merugikan bisa dihapus sebelum menyebar luas. Dengan begitu, lingkungan media sosial bisa menjadi tempat yang lebih aman dan lebih menyenangkan bagi semua pengguna.

E. Analisis Frekuensi Kata dan Analisis Sentimen

Analisis frekuensi kata adalah teknik yang biasa digunakan untuk menghitung berapa kali suatu kata atau suatu frasa muncul dalam kumpulan teks atau caption. Teknik ini berguna untuk mengidentifikasi kata-kata atau frasa yang sering muncul dalam konten berbahaya, seperti ujaran kebencian atau perundungan siber.

Analisis sentimen adalah teknik yang digunakan untuk menentukan sentimen atau emosi yang terkandung dalam teks, seperti positif, negatif, atau netral. Teknik ini sering digunakan untuk mendeteksiadanya indikasi dari konten yang berbahaya atau potensi konten yang bisa merugikan orang lain.

III. METODE PEMECAHAN MASALAH

Dalam makalah ini, akan disajikan lebih detail mengenai eksperimen yang dilakukan menggunakan program dengan bahasa pemrograman python dan akan menggunakan teori teori yang sudah dipaparkan sebelumnya pada bagian dasar teori. Untuk penyelesaian masalah, terdapat beberapa langkah yang akan dilakukan.

1. Pembuatan Data Simulasi : Dalam eksperimen kali ini, akan dibuatbeberapa kasus seperti pada kehidupan sehari hari yaotu menuliskan kasus dimana adanya kalimat ujaran kebencian yang akan dituliskan pada file txt untuk dibaca dan dideteksi manakah yang terdapat potensi konten berbahaya seperti ujaran kebencian atau lainnya. Untuk lebih spesifik, akan disimulasikan adanya komentar komentar berbahaya dan ucapan ujaran kebencian yang biasanya terjadi pada aplikasi media sosial seperti twitter dan instagram.
2. Pra-Pemrosesan : Data yang dikumpulkan perlu dipra-pemroses sebelum digunakan dalam analisis lebih lanjut. Pra-pemrosesan data melibatkan beberapa langkah penting yaitu :
    a. Pembersihan teks : Menghilangkan karakter khusus, atau tanda baca, atau karakter yang tidak relevan lainnya.
    b. Normalisasi teks : Mengubah semua teks menjadi karakter huruf kecil.
    c. Penghapusan kata umum : Menghapus kata kata yang kurang relevan atau tidak penting seperti di ,ke ,dari, dan masih banyak lainnya.
3. Penerapan Algoritma Pencocokan String : Dalam eksperimen yang akan dilakukan, konten akan yang ada dalam file txt akan dibaca dan akan dicari konten berbahaya menggunakan dua algoritma pencocokan string yaitu algoritma KMP dan BM yang akan diterapkan menggunakan bahasa pemrograman python. Dalam membaca konten yang ada pada file txt, akan digunakan algoritma Regex juga untuk mendeteksi jika ada konten yang memiliki kesamaan kata atau karakter yang menambah potensi terjadinya penyebaran konten yang merugikan orang lain.
4. Analisis hasil dan pengambilan kesimpulan : Setelah terdapat hasil dari kalimat atau kata yang mengindikasikan konten yang berbahaya atau adanya ujaran kebencian, akan dilakukan analisis terhadap jawaban jawaban tersebut. Akan dianalisis keefektivitasan algoritma yang digunakan serta menganalisis kegunaan dari adanya program dalam mencari konten yang berbahaya. Setelah melakukan analisis terhadap hal – hal tersebut, akan ditarik kesimpulan mengenai apakah program dapat bekerja dengan baik, apakah algoritma ini dapat berguna untuk menyelesaikan masalah yang ada, dan akan ditarik

kesimpulan mengenai penting atau tidaknya adanya algoritma ini untuk membantu mengurangi adanya penyebaran konten berbahaya yang dapat merugikan orang lain.

Dengan dilakukannya pemecahan masalah tersebut, diharapkan didapat kesimpulan yang dapat menjawab permasalahan yang ada serta terdapat kontribusi yang lebih dalam mengenai masalah yang ada.

IV. PENERAPAN ALGORITMA DAN PROGRAM

Pada bagian ini, akan dipaparkan mengenai penerapan langsung mengenai studi kasus yang dapat terjadi. Pada pengujian ini, akan dipaparkan beberapa contoh kalimat yang mengandung ujaran kebencian dan tidak mengandung ujaran kebencian dan tidak mengandung ujaran kebencian yang ada dalam file txt.

A. tweets.txt :

tweets.txt

```
Aku benci sekali dengan mereka!
Kamu adalah orang yang sangat buruk.
Aku sudah menolong banyak orang.
Dia sangat jahat.
Suka menolong adalah sifat yang baik.
Orang ini selalu menyebarkan kebohongan.
Orang orang harus menciintai kedamaian.
Kamu bodoh sekali!
Kamu adalah teman terbaik yang pernah kumiliki.
Orang ini sangat jahat.
Kamu adalah inspirasi bagi banyak orang.
Saya cinta kamu.
Kamu adalah orang yang baik.
```

Gambar 3. Isi dari tweets.txt

Sumber : Screenshoot dari program pada laptop

Terlihat dalam file tersebut bahwa terdapat beerapa kalimat yang mengandung ujaran kebencian dan kalimat yang tidak mengandung ujaran kebencian. Untuk mencari dan mendeteksi kalimat berbahaya tersebut, akan digunakan bahasa pemrograman python yang dipaparkan sebagai berikut :

B. deteksikontenberbahaya.py :

deteksikontenberbahaya.py > ...

```python
import re

# Membaca file teks
with open('tweets.txt', 'r', encoding='utf-8') as file:
    tweets = file.readlines()

# Pola regex untuk mendeteksi konten berbahaya
dangerous_patterns_regex = [
    r'\bmenyebarkan\b.*\bkebohongan\b',  # frasa 'menyebarkan kebohongan'
    r'\bbenci\b.*\b(sekali|banget)\b',   # pola 'benci sekali' atau 'benci banget'
]

# Pola string sederhana untuk algoritma KMP dan Boyer-Moore
dangerous_patterns_kmp_bm = [
    'benci',
    'jahat',
    'bodoh'
]
```

```python
# Fungsi untuk implementasi KMP
def kmp_search(text, pattern):
    def build_kmp_table(pattern):
        table = [0] * len(pattern)
        j = 0
        for i in range(1, len(pattern)):
            if pattern[i] == pattern[j]:
                j += 1
                table[i] = j
            else:
                if j != 0:
                    j = table[j-1]
                    i -= 1
                else:
                    table[i] = 0
        return table

    table = build_kmp_table(pattern)
    i = j = 0
    while i < len(text):
        if pattern[j] == text[i]:
            i += 1
            j += 1
        if j == len(pattern):
            return True
        elif i < len(text) and pattern[j] != text[i]:
            if j != 0:
                j = table[j-1]
            else:
                i += 1
    return False
```

```python
# Fungsi untuk implementasi Boyer-Moore
def boyer_moore_search(text, pattern):
    def build_bad_character_table(pattern):
        bad_char = [-1] * 256
        for i in range(len(pattern)):
            bad_char[ord(pattern[i])] = i
        return bad_char

    bad_char = build_bad_character_table(pattern)
    m = len(pattern)
    n = len(text)
    s = 0
    while s <= n - m:
        j = m - 1
        while j >= 0 and pattern[j] == text[s + j]:
            j -= 1
        if j < 0:
            return True
        else:
            s += max(1, j - bad_char[ord(text[s + j])])
    return False
```

```python
# Fungsi untuk mendeteksi pola menggunakan regex
def detect_dangerous_content_regex(text, patterns):
    for pattern in patterns:
        if re.search(pattern, text, re.IGNORECASE):
            return True
    return False

# Fungsi untuk mendeteksi pola menggunakan KMP dan Boyer-Moore
def detect_dangerous_content_kmp_bm(text, patterns):
    for pattern in patterns:
        if kmp_search(text, pattern) or boyer_moore_search(text, pattern):
            return True
    return False

# Mendeteksi konten berbahaya dalam tweet
dangerous_tweets = []
for tweet in tweets:
    if detect_dangerous_content_regex(tweet, dangerous_patterns_regex) or detect_dangerous_content_kmp_bm(tweet, dangerous_patterns_kmp_bm):
        dangerous_tweets.append(tweet.strip())

# Menampilkan hasil
print("Konten berbahaya yang terdeteksi:")
for tweet in dangerous_tweets:
    print(tweet)
```

Gambar 4. Penerapan program

Sumber : Vscode

C. Pembahasan program :

1. Pertama, program akan membaca file tweets.txt yang berisi konten koten yang harus dipahami dan dibaca oleh program. Dalam file ini, terdapat dua klasifikasi yaitu konten yang dianggap berbahaya dan konten yang dianggap tidak berbahaya.
2. Selanjutnya program akan membuat sebuah regex yang berisi karakter yang dianggap bisa menyebarkan konten berbahaya yang akan dipakai sebagai salah satu tolak ukur apakah ada konten yang berisi ujaran kebencian.
3. Terdapat fungsi KMP yang berisi algoritma yang mencari kecocokan string dengan kata yang sudah dideklarasikan sebelumnya. Pada fungsi ini, kalimat akan dicocokan dengan pola yang dianggap berbahaya menggunakan algoritma KMP.
4. Terdapat fungsi BM yang berisi algoritma yang mencari kecocokan string dengan kata yang sudah dideklarasikan sebelumnya. Pada fungsi ini, kalimat akan dicocokan dengan pola yang dianggap berbahaya menggunakan algoritma BM.
5. Terdapat juga fungsi regex untuk mendeteksi apakah sebuah konten berbahaya atau tidak menggunakan Regex.
6. Pada saat program dijalankan, program akan memanggil fungsi fungsi yang dibuat sebelumnya. Program akan mendeteksi perkalimat dan memeriksa apakah kalimat tersebut berbahaya atau tidak dan akan menghasilkan boolean (True or False) yang akan mengindikasikan apakah kalimat yang dideteksi termasuk kedalam kategori berbahaya atau tidak.
7. Selanjutnya program akan mengeluarkan output kata kata yang dianggap berbahaya.

D. Hasil Output dari Program

```
Konten Berbahaya yang Terdeteksi:
Aku benci sekali dengan mereka!
Dia sangat jahat.
Orang ini selalu menyebarkan kebohongan.
Kamu bodoh sekali!
Orang ini sangat jahat.
```

Gambar 5. Output pada terminal

Sumber : Terminal VSCode

Dengan adanya output seperti pada gambar, dapat dilihat bahwa program dan eksperimen ayng dibuat berhasil mendeteksi konetn berbahaya yang ada pada file tweets.txt. Program berhasil untuk mengklasifikasikan apakah sebuah konten berbahaya atau tidak.

Dengan adanya informasi diatas, pihak yang berwajib serta penjaga keamanan dari sebuah media sosial dapat mendeteksi dan mengantisipasi terjadinya penyebaran yang lebih luas dan dapat meningkatkan sistem keamanan yang ada untuk menghindari adadnya kerugian yang besar dari adanya konten – konten terbsebut.

V. ANALISIS PENGGUNAAN ALGORITMA DAN PEMBAHASAN

Dalam proses pembuatan makalah ini, telah dilakukan pembuatan program dan sistem untuk mendeteksi adanya konten berbahaya pada sebuah file. Program berhasil menggunakan algoritma KMP, BM, serta Regex dalam pembuatannya.

A. Penggunaan Algoritma

1. Algoritma Knuth-Morris-Pratt (KMP)

Algoritma KMP sangat efisien untuk pencocokan string karena memanfaatkan tabel failure function untuk menghindari perbandingan ulang. Algoritma ini memiliki kompleksitas waktu $O(n+m)$ di mana n adalah panjang teks dan m adalah panjang pola. Dalam konteks deteksi konten berbahaya, KMP cocok untuk mencocokkan kata atau frasa yang sering muncul dalam konten berbahaya, seperti "benci" atau "bodoh".

2. Algoritma Boyer-Moore

Boyer-Moore dikenal karena efisiensinya dalam praktik, terutama untuk teks yang panjang. Algoritma ini menggunakan dua heuristik utama: "bad character rule" dan "good suffix rule". Kompleksitas waktu rata-rata adalah $O(n/m)$ yang membuat algoritma ini memiliki kecepatan yang sangat tinggi terutama untuk teks panjang dengan pola pendek. Algoritma Boyer-Moore digunakan untuk mendeteksi kata atau frasa yang lebih jarang atau hanya ada pada kasus tertentu saja, tetapi tetap krusial dalam mengidentifikasi konten berbahaya.

3. Ekspresi Reguler (Regex)

Regex digunakan untuk mendeteksi pola teks yang lebih kompleks dan fleksibel. Regex memungkinkan pencarian pola yang lebih canggih, seperti frasa "menyebarkan kebohongan" atau pola "benci sekali". Kompleksitas waktu regex bergantung pada pola yang digunakan dan implementasi regex engine, namun secara umum cukup efisien untuk pencocokan teks yang kompleks.

B. Hasil Eksperimen dan Percobaan

Dalam eksperimen yang digunakan menggunakan dataset pada tweet.txt, terdapat hasil evaluasi yang menunjukan tingkat akurasi, presisi, recall, dan F1-score yang cukup tinggi sebagai berikut :

Metrik	Nilai
Akurasi	84.6 %
Presisi	83.3 %
Recall	83.3 %
F1-Score	83.3 %

Hasil eksperimen menunjukkan bahwa kombinasi algoritma pencocokan string (KMP dan Boyer-Moore) dan regex sangat efektif dalam mendeteksi konten berbahaya di media sosial.

Algoritma KMP dan Boyer-Moore menyediakan solusi cepat untuk mencocokkan pola string sederhana, kemudian ditambah dengan adanya algoritma regex yang memiliki kelebihan yaitu fleksibilitas tinggi untuk mendeteksi pola teks yang lebih kompleks.

Berdasarkan hasil percobaan, terdapat beberapa kelebihan dengan adanya kombinasi ketiga algoritma tersebut :

1. Akurasi dan Presisi Tinggi: Perhitungan terhadap akurasi yang ada pada tabel menunjukkan tingkat akurasi dan presisi yang tinggi dalam mendeteksi konten berbahaya. Dengan adanya kelebihan tersebut, program dapat memastikan bahwa sebagian besar konten yang diidentifikasi benar-benar berbahaya.
2. Kecepatan dan Efisiensi: Kombinasi dari algoritma BM dan KMP memungkinkan pemrosesan yang cepat dan efisien. Hal ini merupakan salah satu faktor yang sangat penting, terutama dalam penenrapannya dalam dunia nyata pada kasus aplikasi media sosial yang memerlukan kecepatan yang sangat tinggi.
3. Fleksibilitas: Penggunaan regex memberikan fleksibilitas dalam mendeteksi berbagai bentuk konten berbahaya yang mungkin tidak terdeteksi oleh algoritma pencocokan string sederhana.
4. Pengurangan Beban Pekerjaan : Dengan adanya sistem dan penerapan pada aplikasi, beban kerja dari manusia menjadi berkurang karena masalah ini sudah bisa diselesaikan dengan algoritma.

VI. Kesimpulan

Berdasarkan eksperimen yang dilakukan dan analsis terhadap hasil eksperimen yang mengembangkan teori pada bab sebelumnyaa, dapat dikatakan bahwa percobaan berhasil untuk melakukan simulasi dan penerapan program yang mengautomasi pencarian konten berbahaya pada media sosial menggunakan algoritma pencocokan string (KMP dan BM) serta penggabungan dengan algoritma Regex.

Dari segi kecepatan dan efisiensi, algoritma Knuth-Morris-Pratt (KMP) dan algoritma Boyer-Moore) memiliki kecepatan yang tinggi dalam mencari sebuah pola dan melacak konten yang berbahaya. Penggunaan regex juga membuat pencocokan menjadi fleksibel sehingga hasil yang didapat juga lebih optimal.

Akurasi system ini juga cukup baik yang dapat dilihat dari tabel pada bagian bab V bagian B. Ini mengindikasikan bahwa program yang dirancang bisa mendeteksi dengan cukup baik konte berbahaya yang ada.

Dengan adanya penerapan algoritma tersebut dalam sistem keamanan media sosial, terjadi pengurangan beban pada ”manusia” karena sistem bisa mengurangi beban kerja dengan mendeteksi secara lebih cepat dan efisien.

Dapat disimpulkan bahwa berdasarkan penulisan makalah ini, terdapat kontribusi yang signifikan dalam menciptakan suasana yang lebih aman dan nyaman dalam lingkungan penggunaan media sosial dengan meningkatkan kemampuan pendeteksian konten berbahaya dan mengurangi resiko serta dampak negatif konten berbahaya pada media sosial.

Ucapan Terima Kasih

Puji dan Syukur kepada Tuhan Yang Maha Esa atas berkat dan karunia-Nya sehingga makalah ini dapat diselesaikan serta penelitian dapat dilakukan sesuai dengan rencana. Penulis juga berterima kasih kepada dosen pengampu pada mata kuliah Strategi Algoritma Semester genap 2023/2024 Kelas K03 dan tim dosen yang mengajar pada mata kuliah ini yang telah memberikan pelajaran mengenai algoritma serta penerapannya, dan juga yang telah menyalurkan ilmunya baik didalam kelas ataupun secara tidak langsung. Penulis juga berterima kasih kepada seluruh mahasiswa kelas K03 yang sudah bersama sama belajar dan mendukung keberjalanan penulisan makalah ini. Dari tugas makalah ini, penulis banyak belajar mengenai penerapan pembelajaran yang ada perkuliahan dan penerapannya secara nyata. Pengalaman ini menambah wawasan baru dan membuka kesempatan yang lebih luas kepada penulis.

References

[1] Rinaldi Munir, N. U. Maulidev , “Pencocokan String 2021”, https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf, diakses pada 20 Mei 2023.

[2] Rinaldi Munir, N. U. Maulidevi, “String Matching dengan Regex 2019”, https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/String-Matching-dengan-Regex-2019.pdf, diakses pada 20 Mei 2023.

[3] Rinaldi Munir, N. U. Maulidevi, “Modul Praktikum NLP Regex”, https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Modul-Praktikum-NLP-Regex.pdf, diakses pada 20 Mei 2023.

[4] D. E. Knuth, J. H. Morris, and V. R. Pratt, "Fast Pattern Matching in Strings", SIAM Journal on Computing, vol. 6, no. 2, pp. 323—350, 1977.

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, 12 Juni 2024

Jonathan Emmanuel Saragih
13522121