

# Penggunaan Algoritma Pencocokan String dalam Analisis Sentimen Berita Finansial

Henry Anand Septian Radityo - 13521004

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail (gmail): henryanandseptianradityo@gmail.com

**Abstract**—Berita adalah laporan atau cerita mengenai peristiwa terkini yang disampaikan kepada masyarakat melalui beberapa media seperti surat kabar, televisi, radio dan lainnya. Berita memiliki beberapa topik, salah satu topik yang penting adalah topik mengenai keuangan atau finansial. Berita mengenai finansial merupakan hal yang cukup penting dalam mempengaruhi sentimen masyarakat terhadap suatu topik. Analisis sentimen mengenai berita menjadi hal yang dapat memudahkan untuk dapat memahami tren mengenai keadaan ekonomi. Metode pencocokan string akan menggunakan algoritma pencocokan string terhadap beberapa kata kunci yang sudah disiapkan dan dianggap mewakili sentimen mengenai keadaan finansial. Hasil dari analisis akan menyimpulkan suatu sentimen positif, negatif ataupun netral dari suatu berita.

**Keywords**—finansial; berita; string matching; sentimen;

## I. PENDAHULUAN

Berita adalah suatu laporan mengenai peristiwa yang terjadi di sekitar masyarakat baik yang berada di tingkat lokal, nasional ataupun internasional. Berita biasanya disiarkan melalui media massa seperti koran, majalah, televisi, radio, atau situs web berita. Salah satu jenis dari berita adalah berita finansial atau berita ekonomi, berita finansial adalah jenis berita yang fokus pada perkembangan terbaru dalam dunia keuangan, bisnis, dan ekonomi. Berita finansial biasanya meliputi laporan tentang pergerakan pasar saham, nilai tukar mata uang, harga komoditas, dan berbagai indikator ekonomi lainnya. Berita finansial sangat penting bagi para investor dan pelaku bisnis karena dapat memberikan informasi yang berguna dalam mengambil keputusan investasi dan bisnis.

Algoritma string matching adalah algoritma yang digunakan untuk mencari kemunculan suatu pola atau kata kunci yang berada di dalam sebuah teks. Tujuannya adalah untuk menemukan lokasi kemunculan pola dalam teks. Algoritma ini dapat digunakan dalam berbagai aplikasi, seperti pencarian kata kunci pada mesin pencari, deteksi plagiarisme, dan pemrosesan bahasa alami. Salah satu algoritma string matching yang paling umum digunakan adalah algoritma Knuth-Morris-Pratt (KMP). Algoritma ini bekerja dengan memanfaatkan informasi yang telah diketahui tentang pola yang dicari untuk mempercepat proses pencarian pada teks. KMP memanfaatkan tabel prefix-fungsi (prefix function table) yang dapat dibangun dari pola yang dicari. Tabel ini digunakan untuk menghitung pergeseran yang optimal ketika terjadi

ketidakcocokan antara pola dan teks. Algoritma string matching lainnya yang juga sering digunakan adalah algoritma Boyer-Moore. Algoritma ini bekerja dengan memanfaatkan informasi tentang pola yang dicari dan karakter-karakter pada teks yang telah dicocokkan sebelumnya untuk menentukan pergeseran yang optimal. Algoritma ini dapat bekerja lebih efisien dalam beberapa kasus, terutama ketika pola yang dicari memiliki banyak karakter yang sama di akhir-akhiran pola.

Di era modern, ketersediaan informasi yang sangat banyak membuat tantangan tersendiri untuk dapat menganalisis berita satu persatu. Dalam suatu berita, tersedia beragam informasi mengenai angka – angka dan nama perusahaan serta aksi korporat dan hal lainnya. Dengan memanfaatkan algoritma string matching, pemrosesan data dari suatu berita dapat dipermudah dengan mengetahui pola dari suatu berita atau kata kunci yang ada dalam suatu berita. Dalam konteks finansial, hal ini berhubungan dengan indikator ekonomi, perusahaan, stabilitas keuangan dan kata kunci lain yang berhubungan dengan suatu ekonomi. Penggunaan algoritma pencocokan string dapat membantu melakukan analisis suatu berita secara lebih efisien dan akurat juga mengelompokkan berita menjadi beberapa klasifikasi menurut suatu topik dan kategori tertentu. Makalah ini akan membahas mengenai dasar dari algoritma pencocokan string serta pemanfaatannya dalam melakukan analisis dari berita finansial yang diharapkan hasilnya akan membantu untuk mengetahui informasi mengenai sentimen berita ekonomi.

## II. LANDASAN TEORI

### A. Ekonomi

Ekonomi adalah suatu rumpun ilmu sosial yang mempelajari cara manusia untuk memanfaatkan sumber daya yang ada untuk memenuhi kebutuhannya. Dalam ekonomi terdapat dua jenis ekonomi utama yaitu ekonomi makro dan ekonomi mikro. Ekonomi makro merupakan ekonomi yang mempelajari mengenai ekonomi secara keseluruhan dan berlaku secara luas, seperti inflasi, pertumbuhan ekonomi, krisis moneter, kebijakan ekonomi, dan lainnya. Ekonomi mikro adalah ekonomi yang mempelajari mengenai perilaku dari individu dalam memanfaatkan sumber daya dan pengambilan keputusan untuk melakukan sesuatu. Ekonomi merupakan hal yang sudah ada sejak lama dan memegang peran penting dalam kehidupan manusia.

### B. Berita Ekonomi / Finansial

Berita ekonomi merupakan berita yang menyajikan informasi mengenai ekonomi baik makro maupun mikro. Berita ini juga mencakup berita mengenai keuangan seperti pasar saham, indeks, surat utang dan komoditas. Berita ekonomi sangat penting dalam pelaku ekonomi seperti investor, pengusaha, dan konsumen. Berita ini dapat membantu menyampaikan informasi dan mempengaruhi pelaku ekonomi dalam mengambil keputusan. Berita juga tidak jarang dapat mempengaruhi pasar secara langsung, khususnya berita yang memuat informasi yang cukup penting dengan skala yang besar. Di era digital seperti sekarang, berita ekonomi dapat diakses dengan mudah oleh siapa saja dan dimana saja yang memudahkan pelaku ekonomi untuk menerima informasi dengan cepat.

### C. String

String merupakan salah satu tipe data yang merepresentasikan teks dan kumpulan dari karakter. Karakter dari string dapat berupa huruf, angka, spasi dan tanda baca. String biasa digunakan dalam hal pemrograman dan dituliskan diantara tanda baca petik (“”). Tipe data String juga dapat dilakukan berbagai operasi seperti konkatenasi atau penggabungan string dan pencocokan string dengan menggunakan algoritma – algoritma tertentu.

String juga dapat dipecah menjadi beberapa substrang dan memiliki istilah tertentu yang cukup penting dalam menggunakan algoritma pencocokan string seperti :

1. Prefix : Bagian dari string yang terdiri dari karakter awal pada suatu string.
2. Suffix : Bagian dari string yang terdiri dari karakter akhir pada suatu string.

### D. Algoritma String Matching

Algoritma string matching merupakan algoritma yang mencari kemunculan suatu pola tertentu yang ada pada sebuah teks atau string. Algoritma ini bekerja dengan cara membandingkan karakter yang ada pada satu string dengan sebuah kata kunci yang ingin dicari. Algoritma ini memiliki beberapa pendekatan yang dapat dilakukan seperti penggunaan algoritma brute force, knuth morris pratt, boyer moore, dan Rabin-karp. Setiap algoritma memiliki tingkat kompleksitas tersendiri dan dapat disesuaikan dengan kebutuhan untuk mencapai kinerja yang baik.

### E. Algoritma Brute Force

Algoritma brute force merupakan algoritma paling sederhana dari pencocokan string. Sesuai namanya algoritma ini akan mencari satu persatu karakter dari suatu string pola ke dalam suatu string teks. Algoritma ini merupakan algoritma yang kurang efisien atau memiliki kompleksitas waktu yang tinggi apabila dibandingkan dengan algoritma pencocokan string lainnya dikarenakan tidak menganalisis pola dari suatu string pola. Kompleksitas yang dimiliki oleh algoritma brute force adalah  $O(M*N)$  dengan M adalah panjang pola dan N adalah panjang dari teks string.

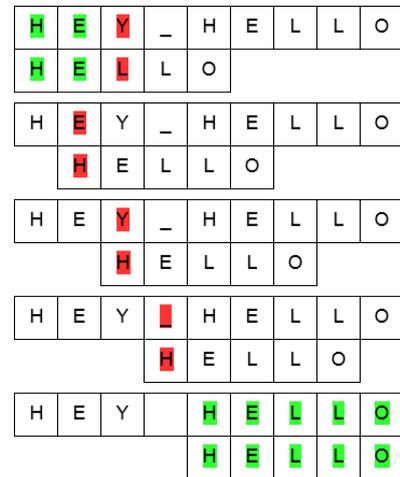


Fig 1. Ilustrasi Algoritma Brute Force

Secara umum algoritma brute force memiliki beberapa langkah sebagai berikut

1. Dari posisi awal string, lakukan iterasi untuk setiap karakter di dalam string.
2. Lakukan perbandingan untuk setiap karakter yang ada di dalam suatu pola string.
3. Jika karakter yang cocok dengan karakter pada pola, maka lanjutkan perbandingan.
4. Jika semua karakter dalam pola cocok, maka pola ditemukan di dalam string.
5. Jika karakter tidak cocok, maka maju ke posisi berikutnya dan ulangi pencocokan karakter.
6. Ulangi hal diatas sampai ditemukannya pola dalam suatu string atau semua karakter pada string sudah dilakukan pencocokan.

### F. Algoritma Knuth-Morris-Pratt

Algoritma Knuth Morris Pratt atau KMP adalah algoritma yang digunakan mencari suatu pola dalam suatu string dengan lebih efisien. Algoritma ini menggunakan cara yang lebih cerdas apabila dibandingkan dengan algoritma brute force karena menghindari perbandingan yang tidak perlu. Algoritma ini memanfaatkan tabel yang biasa disebut dengan border function yang akan menyimpan informasi mengenai string pola yang akan dicari dan bertujuan untuk mempercepat pencarian suatu pola. Algoritma KMP secara umum akan membuat suatu tabel pi terlebih dahulu yang dibentuk dari string pola, dan mulai mencocokkan pola dengan membandingkan karakter pada teks. Jika terjadi ketidakcocokan antar karakter, algoritma akan mencari berapa pergeseran yang harus dilakukan untuk menghindari perbandingan karakter yang tidak perlu sehingga kompleksitas dari algoritma ini menurun. Kompleksitas waktu dari algoritma ini adalah  $O(N)$  dengan N adalah panjang dari teks, algoritma ini sangat efisien apabila digunakan untuk kasus pencarian yang melibatkan perulangan substring pada teks yang akan dicari polanya.

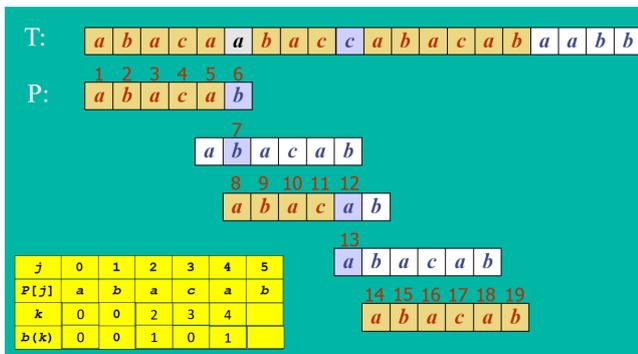


Fig 2. Ilustrasi Algoritma Knuth-Morris-Pratt

Algoritma ini memiliki langkah – langkah umum sebagai berikut :

1. Membuat tabel border function, tabel ini dibuat dengan menginisiasi sebuah array dengan panjang pola string yang kemudian diberi nilai mengenai panjang dari suffix dan prefix yang cocok.
2. Melakukan pencocokan string dengan memperhatikan tabel border function yang akan menentukan besarnya pergeseran.
3. Mengulangi langkah hingga ditemukannya pola atau seluruh karakter di dalam string sudah dilakukan pencarian.

#### G. Algoritma Boyer Moore

Algoritma boyer moore merupakan algoritma pencocokan string yang efisien dan berfokus pada pergeseran secara cerdas. Algoritma ini menggunakan cara pencocokan secara mundur untuk mencocokkan pola terhadap suatu teks. Algoritma ini memiliki keuntungan untuk pencarian pola yang panjang dan kasus banyaknya karakter yang ada pada string tidak cocok sehingga dapat dilewati dengan melakukan pergeseran yang cukup besar.

Secara umum algoritma ini memiliki beberapa langkah yang digunakan untuk mencari pola sebagai berikut

1. Pencocokan dimulai dari karakter pada akhir dari pola dan dilanjutkan dengan karakter yang didepan pola
2. Apabila menemui karakter yang tidak cocok, maka algoritma akan mencari karakter di depan atau di belakang dari pola sehingga memungkinkan untuk melakukan pergeseran.
3. Apabila karakter tidak ada di dalam pola, maka akan dilakukan pergeseran untuk sehingga pola dimulai dari indeks setelah karakter yang tidak cocok.
4. Lakukan perulangan langkah sampai ditemukannya pola dalam sebuah string atau semua karakter telah dilakukan pencocokan.

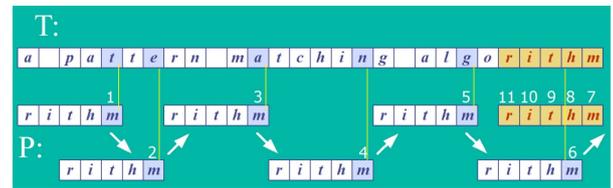


Fig 3. Ilustrasi Algoritma Boyer-Moore

#### H. Data PreProcessing

Preprocessing adalah sebuah langkah yang digunakan untuk membersihkan dan mengorganisir suatu data menjadi bentuk yang lebih bersih sehingga mudah untuk dipahami. Tujuan utama dari preprocessing adalah membuat suatu data lebih mudah untuk dilakukan analisis lebih lanjut. Terdapat beberapa tahapan yang dilakukan pada saat melakukan preprocessing seperti pembersihan data, integrasi data, reduksi data, transformasi data, pemisahan data, dan penanganan data outlier. Tahapan pada preprocessing akan dilakukan sesuai dengan keadaan data dan tujuan dari analisis data.

### III. PEMBAHASAN

Penerapan algoritma pencocokan string dilakukan pada setiap judul dari sebuah berita ekonomi. Pola yang digunakan adalah kata kunci yang dianggap merepresentasikan suatu sentimen terhadap ekonomi. Metode pengambilan judul berita akan menggunakan bantuan dari News API dan Web Scrapping. Proses analisis akan dimulai dengan pengambilan data menggunakan News API dan Scrapping situs – situs berita. Pencarian dengan News API akan difokuskan untuk memperoleh judul berita dengan topik ekonomi dengan memanfaatkan parameter kategori yang sudah tersedia dalam News API. Data yang sudah diambil akan diproses terlebih dahulu sehingga analisis dan data dapat Selanjutnya akan dilakukan penentuan kata kunci dan pencocokan string oleh beberapa algoritma pencocokan string, kali ini metode pencocokan string yang digunakan adalah brute force, Knuth-Morris-Pratt, dan Boyer Moore.

#### A. Pengumpulan Judul Berita Ekonomi – News API

Pada proses pengambilan data, News API menyediakan data dalam bentuk JSON sebagai standar dari pertukaran data yang ada di web. Untuk dapat mengakses data tersebut, dibutuhkan beberapa metode dan teknik sehingga dapat dilakukannya ekstraksi mengenai data dari judul berita saja. Dalam hal ini penulis memilih bahasa pemrograman Python untuk membantu melakukan pengambilan data dalam bentuk JSON. Penulis memanfaatkan library request dalam python untuk melakukan operasi GET pada file json yang sudah tersedia.

Salah satu keunggulan dari News API adalah tersedianya berbagai kategori mengenai berita yang ditampilkan sehingga memudahkan proses pengambilan data. Kategori-kategori tersebut mencakup berita dalam bidang bisnis, teknologi, hiburan, kesehatan, olahraga, sains, politik, travel, seni dan budaya, serta lingkungan. Dalam konteks makalah ini, fokus akan diberikan pada pengambilan dan analisis berita dengan kategori bisnis.

```

import requests
def getTitle():
    api_key =
'de3b0228aaac4cc69bfbfdce911c16985'
    url = f'https://newsapi.org/v2/top-
headlines?country=id&category=business&api
Key={api_key}'
    num_pages = 5

    titles = []
    for page in range(1, num_pages +
1):
        page_url = f'{url}&page={page}'
        response =
requests.get(page_url)

        if response.status_code == 200:
            data = response.json()
            articles = data['articles']

            for article in articles:
                title =
article['title']
                titles.append(title)
            else:
                print('Error occurred:',
response.status_code)

    return titles

```

Fig 4. Pengambilan Judul berita dari News API

### B. Pengumpulan Judul Berita Ekonomi – Web Scrapping

Web scrapping adalah proses ekstraksi data dari suatu halaman dalam website. Web scrapping dapat digunakan untuk mengambil informasi yang relevan dalam halaman suatu website. Untuk melakukan web scrapping, diperlukan pemahaman mengenai tata letak dari suatu halaman website sehingga dapat dilakukan identifikasi bagian – bagian dari judul berita. Setelah mengetahui bagian – bagian dari halaman website, kemudian akan dilakukan ekstraksi dari sebuah HTML menjadi sebuah string yang akhirnya akan dapat diproses. Dalam hal ini, penulis menggunakan bahasa pemrograman Python dengan memanfaatkan library BeautifulSoup sehingga memungkinkan untuk melakukan parsing terhadap suatu HTML dan menemukan elemen judul berita yang nantinya akan dilakukan analisis lebih lanjut.

```

import requests
from bs4 import BeautifulSoup

url =
"https://www.thejakartapost.com/business/e
conomy?page="

list_judul = []

for i in range(1,50):
    response = requests.get(url+str(i))

    soup = BeautifulSoup(response.text,
"html.parser")

```

```

    judul_h2 = soup.find_all("h2")

    for judul in judul_h2:
        list_judul.append(judul.text)

    for el in list_judul:
        print(el)
    with open("list_judul.txt", "w") as
file:
        for judul in list_judul:
            cleaned_judul = judul.strip()
            file.write(cleaned_judul +
"\n")

```

Fig 5. Pengambilan judul berita menggunakan proses Web Scrapping

### C. Penentuan Kata Kunci Sebagai Parameter Sentimen

Pemilihan kata kunci dilakukan dengan mempertimbangkan makna dari suatu kata. Kata – kata positif biasanya akan cenderung merepresentasikan hal yang positif pula seperti “sukses”, “melesat”, “naik” yang dapat merepresentasikan sentimen baik dalam ekonomi. Di sisi lain, kata negatif juga dapat menggambarkan hal yang dianggap merugikan seperti, “gagal”, “kecewa” dan “jatuh”.

Pemilihan kata kunci melibatkan intuisi dan pengalaman serta pola dari suatu data. Pemilihan ini melibatkan unsur subjektif karena mengandalkan intuisi dari manusia. Oleh karena itu, pemilihan ini memerlukan iterasi yang mumpuni agar dapat mengetahui kata kunci yang efektif dalam penggambaran suatu berita.

Dalam hal ini, terdapat suatu library dari nltk yang dapat menentukan kata - kata positif dan juga negatif. Untuk itu penulis menggunakan *library* untuk memudahkan penentuan kata positif dan negatif dari suatu kalimat yang nantinya akan dipadukan dengan menggunakan algoritma *string matching*.

### D. Preprocessing Data

Setelah mendapatkan sebuah data yang diambil dari website. Data tersebut perlu untuk dilakukan pemrosesan ulang sehingga memudahkan proses pencocokan *string*. Pemrosesan data ini akan melibatkan regex dan lemma yang akan menghilangkan semua kata – kata yang kurang bermakna dan membuat semua kata yang ada pada judul menjadi huruf non kapital. Selanjutnya hasilnya akan dituliskan ke dalam *file* txt yang baru dan siap untuk dilakukan pencocokan.

```

import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

def text_transformation(text):
    new_text = re.sub('[^a-zA-Z]', ' ',
text)
    new_text = new_text.lower()
    new_text = new_text.split()
    new_text =
[lemmatizer.lemmatize(word) for word in
new_text if word not in

```

```

set(stopwords.words('english'))
transformed_text = '
'.join(new_text)
return transformed_text

nltk.download('stopwords')
nltk.download('wordnet')
lemmatizer = WordNetLemmatizer()

with open("list_judul.txt", "r") as
file:
    lines = file.readlines()

    corpus = []
    for line in lines:
        transformed_line =
text_transformation(line)
        corpus.append(transformed_line)

    for transformed_text in corpus:
        print(transformed_text)
        with open("transformed_text.txt",
"a") as file:
            file.write(transformed_text +
"\n")

```

Fig 6. Preprocessing data

#### E. String Matching dengan Brute Force

Algoritma brute force bertujuan untuk menganalisis setiap kata yang terdapat dalam array *clean\_titles*. Pembacaan kata akan dilakukan melalui file ekstensi txt yang merupakan file hasil dari preprocessing. Pencocokan string akan melibatkan sebuah library milik corpus yang akan menemukan kata – kata positif dan memasukkannya ke dalam sebuah array.

Setiap judul yang sudah diproses akan di-*split* dan dilakukan pencarian untuk setiap kata di dalam *array* kata positif dan array kata negatif. Setiap kata yang ditemukan di dalam array kata positif akan menambahkan skor sentimen sebanyak satu. Sebaliknya, apabila kata dalam judul ditemukan di dalam *array* negatif, maka sentimen akan berkurang sebanyak satu.

Skor sentimen yang berakhir dengan positif menandakan bahwa suatu judul mengandung sentimen positif dan sebaliknya untuk sentimen yang negatif. Selanjutnya, Hasil dari sentimen beserta judul akan dituliskan ke dalam file txt baru. Penggunaan *library* dan penulisan txt tidak akan disertakan dalam kode dibawah untuk tujuan efisiensi.

```

def driverBF(titles):
    with open("BruteForce.txt", "w") as
output_file:
        for title in titles:
            title = title.replace(", ",
"" ).replace(".", "")
            words = title.split()

            sentiment_score = 0
            for word in words:
                if word in
positive_words:

```

```

                sentiment_score +=
1
            elif word in
negative_words:
                sentiment_score -=
1

            if sentiment_score > 0:
                sentiment = "Positive"
            elif sentiment_score < 0:
                sentiment = "Negative"
            else:
                sentiment = "Neutral"

```

Fig 7. Pencocokan String dengan Brute Force

#### F. String Matching dengan Knuth Morris Pratt

Algoritma Knuth-Morris-Pratt (KMP) digunakan untuk menganalisis setiap kata yang terdapat dalam *array* *clean\_titles*. Pada implementasi ini, kata-kata yang akan dicocokkan dengan kata positif dan kata negatif dalam korpus menggunakan algoritma KMP. Setiap judul dalam *clean\_titles* diproses dengan mengubahnya menjadi huruf kecil dan menghapus tanda baca. Judul kemudian dipecah menjadi kata-kata menggunakan fungsi *split()*. Untuk setiap kata dalam judul, dilakukan pencarian menggunakan algoritma KMP.

Pencocokan string dengan algoritma KMP melibatkan komputasi awal pola (*pattern*) yang disebut dengan fungsi *compute\_prefix*. Fungsi ini menghasilkan nilai prefix yang digunakan untuk mempercepat pencarian string. Setelah komputasi prefix selesai, dilakukan pencarian menggunakan algoritma KMP dengan fungsi *kmp\_search*. Setiap kata dalam judul akan dicocokkan dengan kata positif dalam korpus menggunakan fungsi *kmp\_search*. Jika ada kata positif yang cocok, skor sentimen akan bertambah satu. Selanjutnya, dilakukan pencocokan dengan kata negatif dalam korpus menggunakan fungsi *kmp\_search*. Jika ada kata negatif yang cocok, skor sentimen akan berkurang satu.

Skor sentimen yang berakhir dengan nilai positif menandakan bahwa judul mengandung sentimen positif, sedangkan jika skor sentimen berakhir dengan nilai negatif, maka judul memiliki sentimen negatif. Jika skor sentimen adalah nol, maka judul dianggap netral.

Dalam implementasi KMP ini, langkah-langkah yang dilakukan pada setiap judul hampir sama dengan algoritma brute force, namun menggunakan algoritma KMP untuk pencocokan string yang lebih efisien.

```

def compute_prefix(pattern):
    prefix = [0] * len(pattern)
    j = 0
    for i in range(1, len(pattern)):
        while j > 0 and pattern[i] !=
pattern[j]:
            j = prefix[j - 1]
        if pattern[i] == pattern[j]:
            j += 1
        prefix[i] = j
    return prefix

```

```

def kmp_search(text, pattern):
    prefix = compute_prefix(pattern)
    matches = []
    j = 0
    for i in range(len(text)):
        while j > 0 and text[i] !=
pattern[j]:
            j = prefix[j - 1]
        if text[i] == pattern[j]:
            j += 1
        if j == len(pattern):
            matches.append(i - j + 1)
            j = prefix[j - 1]
    return matches

def driverKMP(titles):
    title_list = []
    sentiment_list = []
    for title in titles:
        title = title.lower()
        title = title.replace(", ",
""").replace(".", "")
        words = title.split()

        sentiment_score = 0
        for word in words:
            if any(kmp_search(word,
positive_word) for positive_word in
positive_words):
                sentiment_score += 1
            elif any(kmp_search(word,
negative_word) for negative_word in
negative_words):
                sentiment_score -= 1

        if sentiment_score > 0:
            sentiment = "Positive"
        elif sentiment_score < 0:
            sentiment = "Negative"
        else:
            sentiment = "Neutral"

```

Fig 8. Pencocokan String dengan Knuth Morris Pratt

### G. String Matching dengan Boyer Moore

Algoritma Boyer-Moore digunakan untuk menganalisis setiap kata yang terdapat dalam array `clean_titles`. Pada implementasi ini, kata-kata akan dicocokkan dengan kata positif dan kata negatif dalam korpus menggunakan algoritma Boyer-Moore. Langkah awal yang dilakukan hampir sama dengan brute force dan KMP yaitu melakukan *preprocessing* data dengan menghapus tanda baca dan kemudian memecah judul menjadi kata-kata menggunakan fungsi `split()`. Untuk setiap kata dalam judul, dilakukan pencarian menggunakan algoritma Boyer-Moore.

Pencocokan string dengan algoritma Boyer-Moore melibatkan komputasi awal pola (*pattern*) yang disebut dengan pembuatan tabel kemunculan terakhir (*last occurrence table*). Tabel ini menyimpan informasi kemunculan terakhir setiap karakter dalam pola, yang akan digunakan untuk mempercepat pencarian *string*.

Setelah tabel kemunculan terakhir dibuat, dilakukan pencarian menggunakan algoritma Boyer-Moore dengan fungsi `boyer_moore_search`. Setiap kata dalam judul akan dicocokkan dengan kata positif dalam korpus menggunakan fungsi `boyer_moore_search`. Jika ada kata positif yang cocok, skor sentimen akan bertambah satu. Selanjutnya, dilakukan pencocokan dengan kata negatif dalam korpus menggunakan fungsi `boyer_moore_search`. Jika ada kata negatif yang cocok, skor sentimen akan berkurang satu.

Penilaian sentimen positif, negatif, dan netral akan menggunakan skor sentimen sebagai penentu. Algoritma boyer moore merupakan algoritma yang lebih efisien dibanding brute force dan untuk beberapa kasus dapat lebih cepat dibanding KMP.

```

def boyer_moore_search(text, pattern):
    m = len(pattern)
    n = len(text)
    if m == 0 or n == 0 or m > n:
        return []

    last_occurrence = {}
    matches = []

    for i in range(m - 1, -1, -1):
        if pattern[i] not in
last_occurrence:
            last_occurrence[pattern[i]]
= i

    i = m - 1
    while i < n:
        j = m - 1
        k = i

        while j >= 0 and text[k] ==
pattern[j]:
            j -= 1
            k -= 1

        if j == -1:
            matches.append(k + 1)

        if text[i] in last_occurrence:
            i += m - min(j, 1 +
last_occurrence[text[i]])
        else:
            i += m

    return matches

def driverBM(titles):
    title_list = []
    sentiment_list = []

    for title in titles:
        title = preprocess_text(title)
        words = title.split()

        sentiment_score = 0
        for word in words:

```

```

if
any(boyer_moore_search(word,
positive_word) for positive_word in
positive_words):
    sentiment_score += 1
elif
any(boyer_moore_search(word,
negative_word) for negative_word in
negative_words):
    sentiment_score -= 1

if sentiment_score > 0:
    sentiment = "Positive"
elif sentiment_score < 0:
    sentiment = "Negative"
else:
    sentiment = "Neutral"

```

Fig 9. Pencocokan String dengan Boyer Moore

#### IV. KESIMPULAN DAN SARAN

Algoritma pencocokan string dapat sangat bermanfaat untuk menentukan secara cepat sentimen dari suatu judul berita. Diantara ketiga jenis algoritma yang digunakan untuk pencocokan string terdapat beberapa kelebihan dan kelemahan untuk masing - masing algoritma.

1. Brute Force : Algoritma mudah untuk diimplementasikan namun memiliki kompleksitas yang tinggi sehingga membuat program berjalan kurang efisien.
2. Knuth Morris Pratt (KMP) : Algoritma ini memiliki border function yang memungkinkan pencocokan menjadi lebih efisien, namun implementasinya lebih sulit dibanding brute force dan untuk kasus tertentu kompleksitasnya dapat sama seperti brute force.
3. Boyer Moore : Algoritma ini memiliki tabel untuk menemukan last occurrence dari sebuah karakter di dalam pattern string. Hal ini membuat boyer moore lebih efisien dibanding brute force dan bagus untuk kasus pola yang sering muncul dalam teks. Namun implementasi dari boyer moore lebih sulit apabila dibandingkan dengan brute force.

Analisis sentimen dengan pencocokan string membutuhkan beberapa peningkatan untuk hal berikut.

1. Pemilihan algoritma yang tepat akan mempercepat pencarian untuk kasus tertentu. Seperti dataset yang besar akan lebih cocok dicari dengan menggunakan boyer moore.
2. Preprocessing data dapat dilakukan dengan lebih dalam seperti mengubah data menjadi bentuk dasarnya sehingga algoritma pencocokan string dapat lebih efisien dalam bekerja.
3. Peningkatan dengan menggunakan machine learning (pembelajaran mesin) akan memberikan hasil yang lebih akurat dibandingkan hanya dengan menggunakan algoritma pencocokan string.

#### VIDEO LINK AT YOUTUBE

Untuk mempermudah penyampaian informasi artikel ini, penulis mencantumkan pranala video YouTube yang menjelaskan dan melakukan demo mengenai algoritma pencocokan string. Pranala tersebut dapat dilihat di : <https://youtu.be/14sNUiaTemo>

#### UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas berkah dan rahmatnya sehingga makalah berjudul "Penggunaan Algoritma Pencocokan String dalam Analisis Sentimen Berita Finansial" ini dapat selesai dengan hasil semaksimal mungkin. Tidak lupa penulis juga mengucapkan terima kasih kepada seluruh dosen pengampu mata kuliah IF2211 Strategi Algoritma yang telah membantu penulis dalam memahami berbagai algoritma termasuk pencocokan string pada makalah ini. Penulis juga mengucapkan terima kasih untuk mahasiswa Teknik Informatika Institut Teknologi Bandung khususnya kelas K3 yang membantu penulis dalam dukungan mental maupun materiil. Tidak lupa penulis juga mengucapkan terima kasih pada orang tua penulis yang selalu mendukung dalam semua yang dilakukan penulis sampai di titik ini.

#### REFERENCES

- [1] Raj, N. (2022). Starters Guide to Sentiment Analysis using Natural Language Processing. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/06/nlp-sentiment-analysis/>
- [2] Liu, B. (n.d.). Opinion Mining, Sentiment Analysis, Opinion Extraction. <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>
- [3] Munir, R. (2023). Pencocokan String. Informatika ITB. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>.
- [4] Munir, R. (2023). Pencocokan String. Informatika ITB. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>

#### 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, 22 Mei 2023



Henry Anand Septian Radityo  
13521004