Aplikasi String Matching dan Regex untuk Identifikasi Topik Populer dalam Analisis Berita di Indonesia

Muhammad Yusuf Rafi- 13522009
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail):

***Abstract*—Informasi mengalir sangat cepat dan melimpah, khususnya berita yang tersebar luas melalui berbagai platform media sehingga menjadi penting untuk mengidentifikasi dan menganalisis tren serta topik yang sedang populer untuk memahami dinamika informasi. Maka dari itu, dibutuhkan Program yang dapat mengidentifikasi tren dan topik populer dalam kumpulan berita. Melalui algortima String Matching dan Regular Expressions (Regex). dapat dihasilkan solusi yang efisien dalam identifikasi topik populer.**

***Keywords*—Tren Berita, String Matching, Regular Expression**

I. Pendahuluan

Dalam era digital yang kita jalani saat ini, laju informasi bergerak dengan kecepatan yang belum pernah terjadi sebelumnya, membanjiri kita dengan berita dari seluruh dunia melalui berbagai platform media digital. Dengan volume berita yang sangat besar ini, kemampuan untuk mengidentifikasi dan menganalisis tren serta topik yang sedang populer menjadi sangat krusial. Hal ini tidak hanya membantu dalam memahami dinamika informasi yang berubah dengan cepat, tetapi juga dalam mengambil keputusan berdasarkan informasi yang relevan dan terkini.

Pengembangan sebuah program yang dapat mengidentifikasi tren dan topik populer dalam kumpulan berita menjadi cukup penting. Adapun, algoritma yang tepat dan efisien, yakni dengan menggunakan algoritma String Matching dan Regular Expressions (Regex). Program ini, dengan menggunakan algoritma pencarian string Boyer-Moore yang terkenal karena keefisienannya, mampu menganalisis teks berita secara cepat untuk mengidentifikasi dan menghitung frekuensi kemunculan kata kunci, juga dengan penggunaan Regex untuk memfilter isi teks paragraf pada HTML dalam suatu berita.Integrasi teknik-teknik ini dalam program tidak hanya mempermudah pengguna dalam menavigasi melalui lautan informasi, tetapi juga memberikan wawasan mendalam mengenai isu-isu yang paling banyak dibicarakan atau diberitakan.

Makalah ini bertujuan untuk menguraikan dan menganalisis bagaimana penerapan algoritma Boyer-Moore dan penggunaan Regex dalam program ini dapat memberikan pemahaman yang lebih baik tentang tren berita saat ini.

II. Landasan Teori

A. *String*

String adalah urutan atau *list* karakter yang digunakan untuk mewakili teks. Karakter dalam string bisa berupa kombinasi dati huruf, angka, simbol, atau karakter khusus lainnya. Misalnya, "hello world", "1234", "#@$!",dan “umur_saya_17!” semuanya adalah contoh string. Dalam banyak bahasa pemrograman, string sudah dijadikan sebagai suatu tipe data bawaan yang memungkinkan pemrogram untuk mendeklarasikan dan mengoperasikan string dengan mudah. Misalnya, dalam Python dan Java, string dapat dideklarasikan dalam tanda kutip ganda seperti `"hello"` atau tanda kutip tunggal seperti `'world'`. Bahasa-bahasa ini juga menyediakan berbagai fungsi dan metode untuk mengolah string, seperti menggabungkan (*concatenation*), pemotongan (*slicing*), pencarian (*searching*), dan penggantian (*replacing*).

B. *String Matching*

String matching adalah proses menemukan keberadaan suatu string (disebut **pola** atau **pattern**) dalam string lain yang lebih panjang (disebut **teks**). Misalnya, mencari kata "main" dalam teks "the rain in spain stays mainly on the plain".

```
T: the rain in spain stays mainly on the plain
P: main
```

Gambar 1. Pencocokan sebuah pattern dalam suatu teks
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

Pada Gambar 1, dilakukan pencocokan *pattern* "main" dalam teks "the rain in spain stays mainly on the plain".

Tentunya, *string matching* ini dapat diimplementasikan dalam banyak hal, dari pencarian dalam editor text, query dalam basis data, analisis urutan genetik, analisis citra, *web*

search engine, dan lainnya.

String Matching memiliki banyak algoritma penyelesaian, diantaranya

1. Brute Force
2. Knutt-Morris-Pratt Algorithm (KMP)
3. Boyer-Moore Algorithm (BM)

C. *Brute Force Algorithm*

Algoritma yang memeriksa semua kemungkinan posisi dalam teks untuk mencocokkan pola yang sesuai. Meskipun paling mudah dipahami, algoritma ini tidak efisien untuk teks yang sangat panjang.

Adapun, tahapan dalam algoritma ini,

1. Ambil posisi pertama di teks dan coba cocokkan dengan huruf pertama pada pola.
2. Pindah ke posisi berikutnya pada teks dan cocokkan kembali dengan huruf berikutnya pada pola dan ulangi proses hingga ditemukan pola yang sesuai pada teks.
3. Jika ada pencocokan yang tidak sesuai, ulangi langkah 1.

```
Teks: NOBODY NOTICED HIM
Pattern: NOT

  NOBODY NOTICED HIM
1 NOT
2  NOT
3   NOT
4    NOT
5     NOT
6      NOT
7       NOT
8        NOT
```

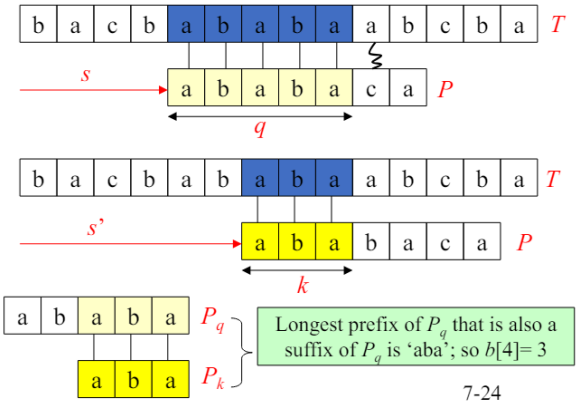
Gambar 2. Ilustrasi String Matching dengan Brute Force
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

D. *Knutt-Morris-Pratt Algorithm*

Algoritma ini mencocokkan pola dalam teks secara berurutan seperti *brute force* dari kiri ke kanan, tetapi, menggeser pola dengan lebih cerdas dengan memanfaatkan informasi yang diperoleh dari pencocokan sebelumnya.

Informasi yang perlu diketahui terlebih dahulu terkait algoritma ini, yakni *prefix* dan *suffix*. Misalkan S adalah sebuah string dengan panjang m, yang dapat dinyatakan sebagai S = $x_0\ x_1\ \ldots\ x_{m-1}$. Dalam konteks ini, *prefix* dari S adalah substring S[0 .. k], yang berarti bagian awal dari string S mulai dari indeks 0 hingga indeks k. Sebaliknya, *suffix* dari S adalah substring S[k .. m-1], yaitu bagian akhir dari string S mulai dari indeks k hingga indeks m-1. Di sini, k adalah indeks yang bisa berada di antara 0 dan m-1. Sebagai contoh, jika S = "abcdefgh", maka beberapa prefix dari S adalah "a", "ab", "abc", dan seterusnya hingga "abcdefgh", sedangkan beberapa suffix dari S adalah "abcdefgh", "bcdefgh", "cdefgh", dan seterusnya hingga "h".

Algoritma *Knuth-Morris-Pratt* menggunakan konsep prefix dan suffix ini untuk membuat tabel LPS (Longest Prefix Suffix). Tabel LPS ini menyimpan panjang dari prefix terpanjang yang juga merupakan suffix untuk setiap posisi dalam pola. Informasi ini nantinya digunakan untuk menggeser pola secara lebih efisien saat terjadi ketidakcocokan selama pencocokan pola dalam teks, sehingga menghindari pemeriksaan yang berulang dan meningkatkan efisiensi algoritma.



Gambar 3. Ilustrasi String Matching dengan KMP
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

Adapun tahapan dalam algoritma KMP,

1. Membangun tabel tabel LPS yang menunjukkan di mana pencocokan sebelumnya gagal.
2. Ambil posisi pertama di teks dan coba cocokkan dengan huruf pertama pada pola.
3. Pindah ke posisi berikutnya pada teks dan cocokkan kembali dengan huruf berikutnya pada pola dan ulangi proses hingga ditemukan pola yang sesuai pada teks.
4. Jika ada pencocokan yang tidak sesuai, gunakan tabel LPS yang menunjukkan pencocokan sebelumnya yang gagal untuk melewatkan karakter yang tidak perlu saat pencocokan dan ulangi langkah 3.

E. *Boyer-Moore Algorithm*

Algoritma ini mencocokkan pola dengan teks dengan menggunakan dua teknik utama (*Looking-Glass Technique* dan *Character-Jump Technique*) untuk mengurangi jumlah perbandingan. ALgoritma ini juga lebih cepar dari *Brute Force* maupun KMP.

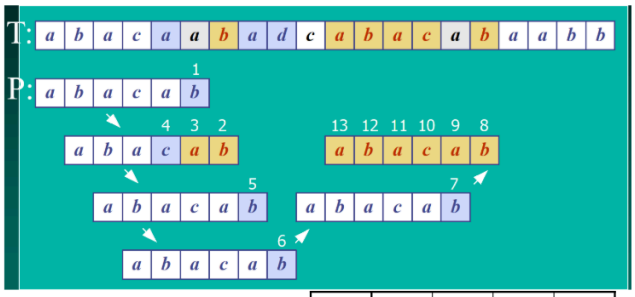
Looking-Glass Technique melakukan perbandingan antara karakter dalam pola dan teks dimulai dari karakter terakhir

pola dan bergerak ke arah awal pola. Ini memungkinkan algoritma untuk memanfaatkan informasi lebih efektif saat terjadi ketidakcocokan, karena informasi di akhir pola bisa langsung digunakan untuk menggeser pola lebih jauh ke kanan dalam teks.

Character-Jump Technique menentukan seberapa jauh pola bisa digeser saat terjadi ketidakcocokan. Saat ketidakcocokan terjadi pada T[i] == x, di mana karakter dalam teks T pada posisi i tidak sama dengan karakter dalam pola P pada posisi j, algoritma menggunakan dua heuristik untuk menentukan pergeseran.

Adapun tahapan dalam algoritma ini,

1. Mulai perbandingan dari karakter terakhir pola dengan karakter teks.
2. Jika terjadi ketidakcocokan dan karakter yang menyebabkan ketidakcocokan ada di pola, geser pola sehingga kemunculan terakhir karakter tersebut di pola sejajar dengan karakter teks yang menyebabkan ketidakcocokan dan ulangi langkah 1.
3. Jika karakter tidak ditemukan dalam pola atau pergeseran berdasarkan karakter buruk tidak memadai, geser posisi pada teks tepat T[i+1] terhadap pola .
4. Jika karakter yang menyebabkan ketidakcocokan tidak ada di pola, geser pola melewati karakter tersebut.



x	a	b	c	d
L(x)	4	5	3	-1

Gambar 4. Ilustrasi String Matching dengan BM
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

F. Regular Expression

Regular expression (Regex) adalah notasi atau format standar yang mendeskripsikan suatu pola (pattern) berupa urutan karakter atau string. Regex digunakan untuk pencocokan string (string matching) dengan efisien. Regex sudah menjadi standar yang tersebar di semua tools.

Regex memiliki banyak notasi dalam identifikasi suatu pola, diantaranya,

Construct	
[abc]	a, b, atau c (simple class)
[^abc]	negasi dari a, b, b
[a-zA-Z]	a sampai z atau A sampai Z, inclusive (range)
[a-d[m-p]]	a sampai d atau m sampai p
[a-z&&[def]]	d, e atau f (irisan)
.	Semua karakter
\d	Digit [0-9]
\D	Non digit [^0-9]
\s	Whitespace character
\S	Non whitespace character
\w	Word character
\W	Non word character
X?	X muncul satu atau tidak sama sekali
X*	X muncul nol atau banyak
X+	X muncul satu atau banyak
x{n}	X muncul tepat n kali

Tabel 1. Beberapa Ekspresi dalam Regex
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Modul-Praktikum-NLP-Regex.pdf

III. METODE

A. Data Berita di Indonesia

Data berita yang digunakan dalam riset ini berasal dari https://newsapi.org/. Melalui tautan tersebut, penulis dapat mengakses API yang menyediakan akses ke berbagai sumber berita. Penggunaan API ini memungkinkan untuk mengambil data berita secara otomatis dari berbagai sumber terpercaya di Indonesia.

Pada riset ini, saya akan mengampbil *top news* dari Indonesia menggunakan API tersebut dengan mendapatkan *key API* penulis terlebih dahulu dengan cara *register* ke API tersebut, lalu menetapkan "Country = id", id menandakan kode negara indonesia. Juga, penulis menetapkan untuk mengambil sebanayk 50 berita saja dengan menetapkan juga "pageSize = 50", pageSize menandakan jumlah berita yang

ingin didapatkan oleh pengguna. Berikut program lengkap penggunaanya dalam bahasa Python,

```python
# API NewsAPI untuk mengambil berita
api_key = '6904129b4fff4bf3a3801c7013c20d33'
url = f'https://newsapi.org/v2/top-headlines?country=id&pageSize=50&apiKey={api_key}'
response = requests.get(url)
data = response.json()
```

Gambar 5. Penggunaan API untuk Pengambilan Berita
Sumber: Hak Cipta Penulis

Pada Gambar 5, setelah mengeksekusi program tersebut, maka akan didapatkan sejumlah artikel, dimana penulis bisa mengakses judulnya, juga tautan menuju berita tersebut.

Kelebihan API ini, berita bisa berasal dari berbagai media di Indonesia, berikut beberapa contoh artikel yang penulis dapatkan.

```
Judul:  Temuan Dinosaurus Baru, Punya Lengan Lebih Kecil dari T-rex - KOMPAS.com
Tautan:  https://news.google.com/rss/articles/CBMicWh0dHBzOi8vd3d3LmtvbXBhcy5jb20vc2FpbnMvcmVhZC8yMDI0LzA2LzExLzE2MzMwMDQyMy
90ZW11YW4tZGlub3NhdXJ1cy1iYXJ1LXB1bnlhLWxlbmdhbi1sZWJpaC1rZWNpbC1kYXJpLXQtcmV40gEA?oc=5
Judul:  Hamas Setujui Resolusi DK PBB soal Gencatan Senjata di Gaza - CNN Indonesia
Tautan:  https://news.google.com/rss/articles/CBMigQFodHRwczovL3d3dy5jbm5pbmRvbmVzaWEuY29tL2ludGVybmFzaW9uYWwvMjAyNDA2MTExNj
A0MzctMTIwLTExMDg1NjUvaGFtYXMtc2V0dWp1aS1yZXNvbHVzaS1kay1wYmItc29hbC1nZW5jYXRhbi1zZW5qYXRhLWRpLWdhemHSAYUBaHR0cHM6Ly93d3cuY2
5uaW5kb25lc2lhLmNvbS9pbnRlcm5hc2lvbmFsLzIwMjQwNjExMTYwNDM3LTEyMC0xMTA4NTY1L2hhbWFzLXNldHVqdWktcmVzb2x1c2ktZGstcGJiLXNvYWwtZ2
VuY2F0YW4tc2VuamF0YS1kaS1nYXphL2FtcA?oc=5
Judul:  Apple Beri Pembaruan App Store, Hadirkan Fitur Nominasi di WWDC 2024 - Gizmologi - Gizmologi.ID
Tautan:  https://news.google.com/rss/articles/CBMiSmh0dHBzOi8vZ2l6bW9sb2dpLmlkL2FwbGlrYXNpL2FwcGxlLWJlcmktcGVtYmFydWFuLWFwcC
1zdG9yZS1kaS13d2RjLTIwMjQv0gFOaHR0cHM6Ly9naXptb2xvZ2kuaWQvYXBsaWthc2kvYXBwbGUtYmVyaS1wZW1iYXJ1YW4tYXBwLXN0b3JlLWRpLXd3ZGMtMj
AyNC9hbXAv?oc=5
Judul:  DPRK desak Pemko serius tangani kasus HIV/AIDS di Banda Aceh - ANTARA Aceh
Tautan:  https://news.google.com/rss/articles/CBMiZmh0dHBzOi8vYWNlaC5hbnRhcmFuZXdzLmNvbS9iZXJpdGEvMzYxMjcyL2RwcmstZGVzYWstcG
Vta28tc2VyaXVzLXRhbmdhbmkta2FzdXMtaGl2LWFpZHMtZGktYmFuZGEtYWNlaNIBamh0dHBzOi8vYWNlaC5hbnRhcmFuZXdzLmNvbS9hbXAvYmVyaXRhLzM2MT
I3Mi9kcHJrLWRlc2FrLXBlbWtvLXNlcml1cy10YW5nYW5pLWthc3VzLWhpdi1haWRzLWRpLWJhbmRhLWFjZWg?oc=5
Judul:  PDIP DKI Sudah Kirim Surat Rekomendasi Cagub Jakarta ke DPP, Ada Nama Anies - detikNews
Tautan:  https://news.google.com/rss/articles/CBMicmh0dHBzOi8vbmV3cy5kZXRpay5jb20vcGVtaWx1L2QtNzM4NTY2MS9wZGlwLWRraS1zdWRhaC
1raXJpbS1zdXJhdC1yZWtvbWVuZGFzaS1jYWd1Yi1qYWthcnRhLWtlLWRwcC1hZGEtbmFtYS1hbmllc9IBdmh0dHBzOi8vbmV3cy5kZXRpay5jb20vcGVtaWx1L2
QtNzM4NTY2MS9wZGlwLWRraS1zdWRhaC1raXJpbS1zdXJhdC1yZWtvbWVuZGFzaS1jYWd1Yi1qYWthcnRhLWtlLWRwcC1hZGEtbmFtYS1hbmllcy9hbXA?oc=5
Judul:  Dokter paparkan tata laksana penanganan pasien osteoporosis - Elshinta.com
Tautan:  https://news.google.com/rss/articles/CBMiZ2h0dHBzOi8vZWxzaGludGEuY29tL25ld3MvMzM4OTU3LzIwMjQvMDYvMTEvZG9rdGVyLXBhcG
Fya2FuLXRhdGEtbGFrc2FuYS1wZW5hbmdhbmFuLXBhc2llbi1vc3Rlb3Bvcm9zaXPSAQA?oc=5
Judul:  Polisi: Ria Ricis Belum Sempat Kirim Rp 300 Juta ke Tersangka Pemerasan - detikNews
Tautan:  https://news.google.com/rss/articles/CBMibmh0dHBzOi8vbmV3cy5kZXRpay5jb20vYmVyaXRhL2QtNzM4NTYxNy9wb2xpc2ktcmlhLXJpY2
lzLWJlbHVtLXNlbXBhdC1raXJpbS1ycC0zMDAtanV0YS1rZS10ZXJzYW5na2EtcGVtZXJhc2Fu0gFyaHR0cHM6Ly9uZXdzLmRldGlrLmNvbS9iZXJpdGEvZC03Mz
g1NjE3L3BvbGlzaS1yaWEtcmljaXMtYmVsdW0tc2VtcGF0LWtpcmltLXJwLTMwMC1qdXRhLWtlLXRlcnNhbmdrYS1wZW1lcmFzYW4vYW1w?oc=5
```

Gambar 6. Kumpulan Berita dari Berbagai Media
Sumber: Hak Cipta Penulis

Pada Gambar 6, didapatkan berita atau artikel dari Kompas, Gizmologi, Antara Aceh, dan detikNews. Tentu masih terdapat banyak artikel lainnya yang berasal dari media lainnya.

B. *Ekstrak Isi Berita*

Setelah mendapatkan kumpulan berita, diperlukan isi dari berita tersebut yang berisikan teks yang cukup panjang. Adapun, penulis menggunakan libary dalam Python, yakni BeautifulSoup untuk pengelolaan tautan berita. BeautifulSoup adalah salah satu library populer yang digunakan untuk mengekstrak data dari dokumen HTML dan XML. Dengan menggunakan BeautifulSoup, penulis dapat dengan mudah menelusuri struktur HTML dari halaman web dan mengekstrak informasi yang dibutuhkan, seperti teks berita dari elemen paragraf (<p>). Berikut program lengkap dalam bahasa Python,

```python
# Fungsi untuk mengekstrak teks dari URL
def extract_text(url):
    try:
        response = requests.get(url)
        response.raise_for_status()
        soup = BeautifulSoup(response.text, 'html.parser')
        paragraphs = soup.find_all('p')
        article_text = ' '.join(p.get_text() for p in paragraphs)
        return article_text
    except (requests.RequestException, ValueError):
        return None
```

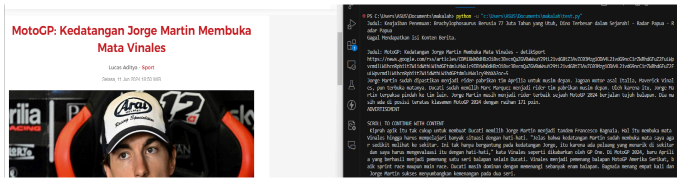
Gambar 6. Penggunaan BeautifulSoupI untuk Ekstrak Isi Berita
Sumber: Hak Cipta Penulis

Selain mengekstrak isi teks pada tautan berita tersebut, penulis juga menggunakan Regex untuk mengelola isi teks berita dengan menghapus semua karakter kecuali huruf (baik huruf besar maupun kecil) dan spasi dari teks dengan menggunakan ekspresi [^a-zA-Z\s]. Setelah itu, semua isi teks diubah ke huruf kecil untuk mempermudah pencocokan string. Berikut program lengkap terkait pembersihan isi teks,

```python
# Fungsi untuk membersihkan teks
def clean_text(text):
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    text = text.lower()
    return text
```

Gambar 7. Penggunaan Regex untuk Pembersihan Isi Teks
Sumber: Hak Cipta Penulis

Setelah, pengeksekusian program pada gambar 6 dan 7, didapatkan isi teks berita secara bersih yang hanya terdiri atas huruf dan spasi. Berikut contoh hasil teks yang didapatkan dari pengolahan tautan berita,



Gambar 8. Isi Teks Berita
Sumber: Hak Cipta Penulis

C. *Implementasi Boyer-Moore*

Implementasi algoritma Boyer-Moore ini untuk pencarian kata yang terdapat pada *keyword* (daftar kata topik) yang terdapat pada teks berita sehingga kata pada *keyword* yang paling sering muncul di beragam berita, maka kata tersebut yang sedang menjadi topik populer atau tren pada berita saat ini.

Penulis telah mendefinisikan *keyword* dengan daftar string, yakni "ekonomi", "kesehatan", "pemerintahan", "teknologi",

"olahraga", "pendidikan", "budaya", "lingkungan", "politik", "hukum","agama", "infrastruktur", "sosial", "internasional", "keuangan", "investasi", "bisnis", "seni", "hiburan", "pariwisata", "pertanian", "perikanan", "kriminalitas". Kata-kata tersebut merupakan daftar topik berita di Indonesia.

Setelah mendefinisikan *keyword*, dibuatkan implementasi algoritma Boyer-Moore dengan modifikasi dimana fungsi BM ini akan *return* jumlah kemunculan suatu kata dalam teks, bukan indeks pertama pola dalam teks. Berikut tahapannya,

1. Membangun tabel kemunculan terakhir dari pola.
2. Memulai pencarian dari akhir teks dan akhir pola.
3. Melakukan pencocokan karakter di akhir pola dengan karakter di teks.
4. Jika karakter cocok, kita mundur ke belakang untuk memeriksa karakter sebelumnya.
5. Jika tidak cocok, kita menggunakan informasi dari tabel kemunculan terakhir untuk memutuskan perpindahan karakter pola.
6. Ulangi langkah 2-5 hingga pola ditemukan dan mencapai akhir teks dan mengembalikan jumlah kemunculan pola dalam teks.

Berikut implementasi programnya dalam bahasa Python,

```python
# Fungsi untuk membuat tabel kemunculan terakhir
def buildLast(pattern):
    last = [-1] * 128
    for i in range(len(pattern)):
        last[ord(pattern[i])] = i
    return last

# Fungsi pencarian Boyer-Moore
def boyer_moore_search(text, pattern):
    last = buildLast(pattern)
    n = len(text)
    m = len(pattern)
    i = m - 1
    if i > n - 1:
        return 0  # tidak ada pencocokan jika pola lebih panjang dari teks

    j = m - 1
    count = 0
    while i <= n - 1:
        if pattern[j] == text[i]:
            if j == 0:
                count += 1  # pola cocok, tambahkan ke jumlah
                i += m  # lompat ke depan m karakter
                j = m - 1
            else:
                i -= 1
                j -= 1
        else:
            lo = last[ord(text[i])] if ord(text[i]) < 128 else -1  # kemunculan terakhir
            i = i + m - min(j, 1 + lo)
            j = m - 1

    return count
```

Gambar 9. Implementasi Boyer-Moore
Sumber: Hak Cipta Penulis

D. *Pengumpulan Semua Berita*

Setelah menyiapkan fungsi yang diperlukan, maka diperlukan pengumpulan semua berita yang telah didapatkan dari API ke dalam suatu variabel. Penulis telah membuat variabel bernama “articles” yang terdiri atas 2 data pada masing-masing elemennya, yakni judul dari berita dan tautan dari berita. Hal ini tentunya sangat krusial karena penulis menghindari akses API secara berulang, melainkan mengakses API sekali dan sudah menyimpan semua daftar berita yang didapatkan.

Setelah itu, diperlukan juga penghitung frekuensi kata kunci dari artikel pertama hingga artikel terakhir. Penulis telah membuat juga variabel bernama ‘total_word_frequencies’ yang berisikan data kemunculan *word* dalam *keyword* dengan bentuk *integer*. Awalnya, isi dari total total_word_frequencies akan ditetapkan dengan nilai 0, lalu seiring dengan pengolahan berita, pencocokkan *word* pada *keyword* di teks berita(yang sudah diolah dengan Regex) dengan Boyer-Moore, maka nilai-nilai pada variabel penghitung frekuensi ini akan bertambah sesuai nilai frekuensi kata pada teks berita tersebut. Berikut Program lengkapnya dalam bahasa Python,

```python
articles = []
for article in data['articles']:
    articles.append((article['title'], article['url']))


# Inisialisasi total word frequencies
total_word_frequencies = {word: 0 for word in keyword}

# Menganalisis beberapa artikel pertama
for title, article_url in articles:
    print(f"Judul: {title}")
    article_text = extract_text(article_url)
    if article_text:
        cleaned_text = clean_text(article_text)
        word_frequencies = {word: boyer_moore_search(cleaned_text, word) for word in keyword}
        print(f"Jumlah Kemunculan Kata pada Article: {word_frequencies}\n")

        # Menambahkan frekuensi kata kunci dari artikel saat ini ke total_word_frequencies
        for word, freq in word_frequencies.items():
            total_word_frequencies[word] += freq
    else:
        print("Gagal Mendapatkan isi Konten Berita.\n")

# Menampilkan total frekuensi kata kunci di semua artikel
print("Total Kemunculan Kata pada semua Article:")
for word, freq in total_word_frequencies.items():
    print(f"{word}: {freq}")
```

Gambar 10. Proses Pengolahan Semua Berita
Sumber: Hak Cipta Penulis

Pada Gambar 10, program menampilkan jumlah frekuensi kata di tiap artikel sehingga pengguna bisa memeriksa jumlah kemunculan frekuensi kata di tiap artikel, dimana kata dengan frekuensi tertinggi menentukan topik dari berita tersebut dan di akhir program akan ditampilkan semua *word* pada *keyword* beserta jumlah frekuensinya sehingga word dengan jumlah frekuensi tertinggi menandakan topik tersebut yang sedang tren di berita di Indonesia saat ini.

E. *Pengujian*

Pengujian pertama dilakukan dengan mengambil berita terkini di Indonesia dari API sejumlah 50. Berikut hasil keluaran yang didapatkan,

```
Judul: Nilai Tukar Rupiah terhadap Dolar AS Hari Ini, Selasa (11/6/2024) - Bisnis.com
Jumlah Kemunculan Kata pada Article: {'ekonomi': 1, 'kesehatan': 0, 'pemerintahan': 0, 'teknologi': 0, 'olahraga': 0, 'pendidikan': 0, 'budaya': 0, 'lingkungan': 0, 'politik': 1, 'hukum': 0, 'agama': 0, 'infrastruktur': 0, 'sosial': 0, 'internasional': 0, 'keuangan': 1, 'investasi': 0, 'bisnis': 9, 'seni': 4, 'hiburan': 0, 'pariwisata': 0, 'pertanian': 0, 'perikanan': 0, 'kriminalitas': 0}

Judul: Peneliti: Air Beku Terdeteksi di Gunung Tertinggi Planet Mars - KOMPAS.com
Jumlah Kemunculan Kata pada Article: {'ekonomi': 0, 'kesehatan': 0, 'pemerintahan': 0, 'teknologi': 0, 'olahraga': 0, 'pendidikan': 0, 'budaya': 0, 'lingkungan': 0, 'politik': 0, 'hukum': 0, 'agama': 0, 'infrastruktur': 0, 'sosial': 0, 'internasional': 1, 'keuangan': 0, 'investasi': 0, 'bisnis': 0, 'seni': 2, 'hiburan': 0, 'pariwisata': 0, 'pertanian': 0, 'perikanan': 0, 'kriminalitas': 0}
```

Gambar 11. Keluaran Beberapa Artikel beserta Frekuensi Kemunculan Kata
Sumber: Hak Cipta Penulis

Pada gambar 11, didapatkan frekuensi kemunculan kata dari artikel berjudul Nilai Tukar Rupiah terhadap Dolar AS Hari Ini dari https://bisnis.com/ , didapatkan kata dengan frekuensi tertinggi, yakni 'Bisnis' sejumlah 9. Ini menandakan bahwa memang benar artikel ini mengandung topik bisnis. Program ini lalu dilanjutkan dengan keluaran akhir sebagai berikut.

```
Total Kemunculan Kata pada semua Article:
ekonomi: 1
kesehatan: 7
pemerintahan: 1
teknologi: 12
olahraga: 0
pendidikan: 0
budaya: 1
lingkungan: 0
politik: 4
hukum: 15
agama: 5
infrastruktur: 0
sosial: 2
internasional: 2
keuangan: 7
investasi: 10
bisnis: 13
seni: 18
hiburan: 0
pariwisata: 0
pertanian: 0
perikanan: 0
kriminalitas: 0
```

Gambar 12. Keluaran Beberapa Frekuensi Jumlah Kemunculan Kata dari Seluruh Artukel
Sumber: Hak Cipta Penulis

Pada gambar 12, dihasilkan keluaran berupa frekuensi total jumlah kemunculan *word* pada *keyword* dari seluruh artikel berita. didapatkan frekuensi tertinggi, yakni 'seni' dengan nilai 18, ini menandakan topik yang sedang tren saat ini adalah seni. Adapun berita terkait topik lainnya, yakni ada terkait ekonomi, kesehatan, pemerintahan, teknologi, hukum, politik, agama, internasional, dan keuangan.

IV. Pembahasan

Dalam riset ini, penulis menggunakan data berita dari https://newsapi.org/ untuk menganalisis topik-topik yang sedang tren di Indonesia. Penggunaan API tersebut memungkinkan penulis untuk mengambil data berita dari berbagai sumber terpercaya di Indonesia. Dengan menetapkan parameter seperti "Country = id" dan "pageSize = 50", penulis bisa mendapatkan top news dari Indonesia dalam jumlah maksimal 50 berita.

Untuk mengambil isi berita, penulis menggunakan BeautifulSoup dalam Python untuk mengelola tautan berita dalam mengolah struktur HTML. Dengan demikian, dapat diekstrak teks berita dari elemen paragraf (<p>). Selain itu, dilakukan pembersihan teks menggunakan Regex untuk menghapus karakter selain huruf dan spasi, serta mengubah semua huruf menjadi huruf kecil.

Setelah itu, diimplementasikan algoritma Boyer-Moore untuk mencari kata-kata kunci yang muncul dalam berita. Hal ini dilakukan untuk menentukan topik yang sedang populer atau tren dalam berita. Sebuah daftar kata kunci telah ditetapkan, dan algoritma Boyer-Moore digunakan untuk menghitung frekuensi kemunculan kata-kata kunci tersebut dalam teks berita. Setelah mengumpulkan berita, penulis menghitung frekuensi kemunculan kata kunci dari setiap artikel. Hal ini dilakukan untuk mengetahui topik apa yang paling banyak dibahas dalam berita tersebut. Hasilnya kemudian digunakan untuk menentukan topik yang sedang tren saat itu.

Pengujian dilakukan dengan mengambil 50 berita terkini dari Indonesia. Dari hasil pengujian, dapat dilihat frekuensi kemunculan kata-kata kunci dalam berbagai artikel, dan dari situ dapat diketahui topik yang sedang tren saat itu. Waktu eksekusi program cukup cepat dengan rata-rata waktu eksekusi 2-3 menit secara keseluruhan, adapun untuk waktu terkait pencocokan Boyer-Moore pada suatu artikel memiliki rata-rata 5-7 detik. waktu eksekusi ini cukup lama secara keseluruhan karena diperlukannya pencocokan string pada setiap artikel dan tiap artikel memiliki isi teks yang panjang. Tentunya, dengan jumlah artikel yang lebih sedikit, bisa mempercepat waktu eksekusi. Selain itu, pemilihan algoritma Boyer-Moore menurut penulis sudah cukup tepat karena secara kompleksitas waktu, rata-rata waktunya O(n/m), di mana n adalah panjang teks dan m adalah panjang pola. Dalam kasus terburuk, kompleksitas waktu bisa mencapai O(n), tetapi ini jarang terjadi

Adapun untuk optimalisasi program, bisa dengan mengimplementasikan *parallel processing* atau multiprocessing untuk memproses beberapa artikel secara bersamaan tanpa menunggu sesuai urutan.

V. Kesimpulan

Melalui eksplorasi dalam penerapan *String Matching* dan *Regex* dalam analisis tren dan topik populer di Indonesia, penelitian ini memberikan hasil yang cukup memuaskan. Riset ini berhasil memberikan solusi yang efisien dan efektif dalam mencari tren dan topik populer dalam beragam berita di Indonesia

Selain itu, penelitian ini tentunya bisa juga diterapkan dengan algoritma lainnya selain *Boyer-Moore karena* dalam penyelesaian *String Matching*, terdapat banyak sekali penyelesaian lainnya yang bisa memiliki solusi yang lebih efisien. Namun, yang diprioritaskan tetaplah hasil solusi yang optimal dan efisien.

VI. Ucapan Terima kasih

Pertama-tama, Saya ucapkan rasa syukur kepada Tuhan Yang Maha Esa atas berkat-Nya dan keberkahan-Nya saya bisa menyelesaikan penulisan makalah ini. Tidak lupa juga, Saya ucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi, M.T., atas bimbingannya selama pembelajaran mata kuliah IF2211 Strategi Algoritma. Tentunya, penyelesaian makalah ini bukan karena usaha Saya sendiri, tetapi juga pihak-pihak lain yang telah mendukung saya selama penulisan makalah. Terakhir, Saya berharap penelitian ini bisa bermanfaat untuk para pembaca terutama dalam menganalisis tren serta topik yang sedang populer karena hal ini tidak hanya membantu dalam memahami dinamika informasi yang berubah dengan cepat, tetapi juga dalam mengambil keputusan berdasarkan informasi yang relevan dan terkini.

References

[1] Rinaldi Munir, "Pencocokan string (String matching/pattern matching) " 2024.[Online].Available:https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf [Diakses pada 12 Juni 2024]

[2] Rinaldi.Munir,"ModulPraktikumRegex".[Online]. Available:https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Modul-Praktikum-NLP-Regex.pdf [Diakses pada 12 Juni 2024]

[3] Rinaldi Munir, "Pencocokan string dengan Regular Expression " 2024.[Online].Available:https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/String-Matching-dengan-Regex-2019.pdf [Diakses pada 12 Juni 2024]

[4]

[5] GeeksforGeeks, "Algoritma Boyer-Moore untuk Pencarian Pola".[Online].Available:https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/ . [Diakses pada 12 juni 2024].

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

Muhammad Yusuf Rafi/13522009