LAPORAN
TUGAS KECIL 4 IF2211 STRATEGI ALGORITMA
Ekstraksi Informasi dari Artikel Berita dengan Algoritma Pencocokan String

Disusun oleh:
13518117 – Muhammad Firas

TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2020

BAB I
DESKRIPSI TUGAS

Algoritma pencocokan string (pattern) Knuth-Morris-Pratt (KMP) dan Algoritma Boyer-Moore merupakan algoritma yang lebih baik daripada brute force. Pada Tugas Kecil IV kali ini Anda diminta membuat aplikasi sederhana ekstraksi informasi dengan kedua algoritma tersebut, plus menggunakan regular expression (regex). Teks yang akan Anda proses adalah teks berita berbahasa Indonesia seperti contoh berikut ini (jabar11042020.txt).

421 Orang di Jabar Terkonfirmasi Positif COVID-19
Yudha Maulana - detikNews
Sabtu, 11 Apr 2020 20:07 WIB
Bandung - Angka positif virus Corona atau COVID-19 di Jawa Barat menembus angka 400 kasus. Laman Pusat Informasi dan Koordinasi COVID-19 Jabar (Pikobar) pada Sabtu (11/4/2020) pukul 18.43 WIB, mencatat terdapat 421 orang yang terkonfirmasi positif COVID-19.

Dibandingkan sehari sebelumnya, jumlah tercatat yaitu 388 orang. Terjadi penambahan 8,5 persen atau 33 kasus per harinya. Sementara itu, secara nasional terdapat 3.842 kasus positif COVID-19.
Dari 421 kasus tersebut, 40 orang meninggal dunia dengan keterangan terpapar COVID-19. Sedangkan, angka kesembuhan di Jabar masih tetap berada di angka 19 orang.
Per hari jumlah Orang Dalam Pemantauan (ODP) di Jabar mencapai 28.775 orang. Sebanyak 15.363 di antaranya masih menjalani proses pemantauan dan 13.412 orang lainnya telah selesai menjalani proses pemantauan.
Sementara itu jumlah Pasien Dalam Pengawasan (PDP) mencapai 2.278 orang. Tercatat 1.344 orang masih menjalani proses pengawasan dan 934 orang lainnya telah selesai menjalani proses pengawasan.

Pada kumpulan teks berita korban covid-19 ini, informasi penting dari pengguna adalah jumlah korban dan waktunya. Oleh karena itu, informasi yang akan diekstraksi adalah angka (diberi warna biru) dan waktu (diberi warna merah).

Pengguna aplikasi ini akan memberikan masukan berupa folder yang berisi kumpulan teks berita, keywords, dan hasil ekstraksi jumlah dan waktunya. Karena sebagian besar kalimat mengandung angka, aplikasi akan memfilter angka berdasarkan keywords dari pengguna, seperti ‘terkonfirmasi positif’, ‘meninggal dunia’, ‘Orang Dalam Pemantauan’, ‘ODP’, ‘Pasien Dalam Pengawasan’, ‘PDP’ atau keyword lainnya. Hasilnya berupa pasangan angka dan waktu, serta kalimat yang mengandung informasi tersebut. Waktu yang diambil harus berada dalam satu kalimat dengan angka tersebut. Jika tidak ada, gunakan tanggal artikel yang tercantum. Jika terdapat lebih dari satu angka, pilih angka yang paling dekat dengan keyword. Berikut contohnya.

> Keyword: *terkonfirmasi positif*
> Hasil ekstraksi informasi:
> Jumlah: *421;* Waktu: Sabtu, 11 Apr 2020 20:07 WIB
> 421 Orang di Jabar **Terkonfirmasi Positif** COVID-19. (jabar11042020.txt)

Jumlah: *421;* Waktu: *Sabtu (11/4/2020) pukul 18.43 WIB*
Laman Pusat Informasi dan Koordinasi COVID-19 Jabar (Pikobar) pada Sabtu (11/4/2020) pukul 18.43 WIB, mencatat terdapat 421 orang yang **terkonfirmasi positif** COVID-19. (jabar11042020.txt)

Keyword: *meninggal dunia*
Hasil ekstraksi informasi:
Jumlah: *40;* Waktu: Sabtu, 11 Apr 2020 20:07 WIB
Dari 421 kasus tersebut, 40 orang **meninggal dunia** dengan keterangan terpapar COVID-19. (jabar11042020.txt)

Terdapat dua jenis pencocokan string yang Anda lakukan. Pertama, exact match dengan keyword yang diberikan pengguna untuk memfilter kalimat yang akan diproses informasinya. Semua teknik (KMP, BM, dan regex) bisa digunakan untuk fitur ini. Kedua, ekstraksi jumlah dan waktu dari kalimat hasil exact match dengan menggunakan regex.

Pencarian tidak bersifat *case sensitive*, jadi huruf besar dan huruf kecil dianggap sama (hal ini dapat dilakukan dengan mengganggap seluruh karakter di dalam pattern dan teks sebagai huruf kecil semua atau huruf kapital semua).

BAB II
TEORI DASAR

2.1. Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt adalah salah satu algoritma pencarian string, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977.

Jika kita melihat algoritma brute force lebih mendalam, kita mengetahui bahwa dengan mengingat beberapa perbandingan yang dilakukan sebelumnya kita dapat meningkatkan besar pergeseran yang dilakukan. Hal ini akan menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pencarian.

Secara sistematis, langkah-langkah yang dilakukan algoritma Knuth-Morris-Pratt pada saat mencocokkan string:

1. Algoritma Knuth-Morris-Pratt mulai mencocokkan pattern pada awal teks.
2. Dari kiri ke kanan, algoritme ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
    1. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
    2. Semua karakter di pattern cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser pattern berdasarkan tabel next, lalu mengulangi langkah 2 sampai pattern berada di ujung teks.

2.2. Algoritma Boyer-Moore

Algoritma Boyer-Moore adalah salah satu algoritme pencarian string, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977.

Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian string yang ditemukan sebelumnya, algoritma Boyer-Moore mulai mencocokkan karakter dari sebelah kanan pattern. Ide di balik algoritma ini adalah bahwa dengan memulai pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat.

Secara sistematis, langkah-langkah yang dilakukan algoritme Boyer-Moore pada saat mencocokkan string adalah:

1. Algoritme Boyer-Moore mulai mencocokkan pattern pada awal teks.
2. Dari kanan ke kiri, algoritme ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
    1. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
    2. Semua karakter di pattern cocok. Kemudian algoritme akan memberitahukan penemuan di posisi ini.

3. Algoritme kemudian menggeser pattern dengan memaksimalkan nilai penggeseran good-suffix dan penggeseran bad-character, lalu mengulangi langkah 2 sampai pattern berada di ujung teks.

2.3 Regular Expression

Regular expression didefinisikan berdasarkan aturan teori bahasa formal. Regular expression terdiri dari konstanta dan operator yang menunjukkan himpunan-himpunan string dan operasi antar himpunan string tersebut secara berurutan.

Konstanta yang telah didefinisikan adalah:

- Himpunan kosong, diberi notasi ∅.
- String kosong, diberi notasi ε.
- Karakter, diberi notasi sesuai dengan karakter bahasa yang digunakan.

Operator yang telah didefinisikan adalah:

- Konkatenasi, misal {"ab", "c"} {"d", "ef"} = {"abd", "abef", "cd", "cef"}.
- Alternasi, misal {"ab", "c"} | {"ab", "d", "ef"} = {"ab", "c", "d", "ef"}.
- Kleene star, menunjukkan semua himpunan yang dapat dibuat dengan melakukan konkatenasi 0 atau lebih banyak string dari string yang dilakukan operasi ini. Misal {"ab", "c"}* = {ε, "ab", "c", "abab", "abc", "cab", "cc", "abababab", "abcab", ... }.

BAB III
IMPLEMENTASI

Algoritma Knuth-Morris-Pratt

```
def kmpAlgorithm(text, pattern, fail):
    lenText = len(text)
    lenPat = len(pattern)

    i = 0
    j = 0

    while (i < lenText):
        if(pattern[j] == text[i]):
            if(j == lenPat - 1):
                return (i - lenPat + 1)
            i += 1
            j += 1
        elif(j > 0):
            j = fail[j-1]
        else:
            i += 1
    return -1


def matchFail(pattern):
    fail = [0]*len(pattern)
    fail[0]

    j = 0
    i = 1

    while (i < len(pattern)):
        if (pattern[i] == pattern[j]):
            fail[i] = j + 1
            j += 1
            i += 1
        elif (j > 0):
            j = fail[j-1]
        else:
            fail[i] = 0
            i += 1
    return fail
```

Algoritma Boyer-Moore

```
def bmAlgo(text, pattern, last):
    lenText = len(text)
    lenPat = len(pattern)
    i = lenPat-1

    if(i > lenText-1):
        return -1

    j = lenPat-1
    while(True):
        if(pattern[j] == text[i]):
            if(j == 0):
                return i
            else:
                i -= 1
                j -= 1
        else:
            lastOcc = last[ord(text[i])]
            i = i + lenPat - min(j, 1+lastOcc)
            j = lenPat - 1
        if(i > lenText-1):
            break
    return -1

def buildLast(pattern):
    last = [-1]*128

    for i in range(len(pattern)):
        last[ord(pattern[i])] = i

    return last
```

Regular Expression

```
def regex(pattern, text):
    arrSentence = re.findall(re.compile(r'(?:^|[\.\n] )(.*?' + pattern + r'.*?
)(?=[\.\n])', re.I), text)
    return arrSentence
```

BAB IV
UJI KASUS

- Screenshot input-output program untuk algoritma Knuth-Morris-Pratt

```
D:\Document Firas\Coolyeah\Semester 4\Stima\Tucil 4>C:/Users/ASUS/AppData/Local/Programs/Python/Python37/python.exe "d:/Document Firas/Coolyeah/
Semester 4/Stima/Tucil 4/KMPAlgo.py"
Masukkan directory: test
Masukkan pencarian: terkonfirmasi positif
Masukkan algoritma pencarian:
1. Knuth-Morris-Pratt
2. Boyer-Moore
3. Regular Expression
1
Hasil ekstraksi:
Jumlah:  7.135 Waktu: Selasa (21/4/2020)
 "Kasus terkonfirmasi positif COVID-19 totalnya menjadi 7.135 orang," kata Juru Bicara Pemerintah untuk Penanganan wabah Virus Corona, dr Achmad
 Yurianto, dalam konferensi pers yang ditayangkan saluran YouTube Badan Nasional Penanggulangan Bencana (BNPB), Selasa (21/4/2020).
(test\detik1.txt)

Jumlah: 421  Waktu:
 421 Orang di Jabar Terkonfirmasi Positif COVID-19
(test\jabar11042020.txt)

Jumlah:  421  Waktu: Sabtu (11/4/2020)
  Laman Pusat Informasi dan Koordinasi COVID-19 Jabar (Pikobar) pada Sabtu (11/4/2020) pukul 18.43 WIB, mencatat terdapat 421 orang yang terkonf
irmasi positif COVID-19.
(test\jabar11042020.txt)
```

- Screenshot input-output program untuk algoritma Boyer-Moore

```
D:\Document Firas\Coolyeah\Semester 4\Stima\Tucil 4>C:/Users/ASUS/AppData/Local/Programs/Python/Python37/python.exe "d:/Document Firas/Coolyeah/
Semester 4/Stima/Tucil 4/KMPAlgo.py"
Masukkan directory: test
Masukkan pencarian: terkonfirmasi positif
Masukkan algoritma pencarian:
1. Knuth-Morris-Pratt
2. Boyer-Moore
3. Regular Expression
2
Hasil ekstraksi:
Jumlah:  7.135 Waktu: Selasa (21/4/2020)
 "Kasus terkonfirmasi positif COVID-19 totalnya menjadi 7.135 orang," kata Juru Bicara Pemerintah untuk Penanganan wabah Virus Corona, dr Achmad
 Yurianto, dalam konferensi pers yang ditayangkan saluran YouTube Badan Nasional Penanggulangan Bencana (BNPB), Selasa (21/4/2020).
(test\detik1.txt)

Jumlah: 421  Waktu:
 421 Orang di Jabar Terkonfirmasi Positif COVID-19
(test\jabar11042020.txt)

Jumlah:  421  Waktu: Sabtu (11/4/2020)
  Laman Pusat Informasi dan Koordinasi COVID-19 Jabar (Pikobar) pada Sabtu (11/4/2020) pukul 18.43 WIB, mencatat terdapat 421 orang yang terkonf
irmasi positif COVID-19.
(test\jabar11042020.txt)
```

- Screenshot input-output program untuk Regular Expression

```
D:\Document Firas\Coolyeah\Semester 4\Stima\Tucil 4>C:/Users/ASUS/AppData/Local/Programs/Python/Python37/python.exe "d:/Document Firas/Coolyeah/
Semester 4/Stima/Tucil 4/KMPAlgo.py"
Masukkan directory: test
Masukkan pencarian: terkonfirmasi positif
Masukkan algoritma pencarian:
1. Knuth-Morris-Pratt
2. Boyer-Moore
3. Regular Expression
3
Hasil ekstraksi:
Jumlah: 421  Waktu:
 421 Orang di Jabar Terkonfirmasi Positif COVID-19(test\jabar11042020.txt)

Jumlah:  421  Waktu: Sabtu (11/4/2020)
 Laman Pusat Informasi dan Koordinasi COVID-19 Jabar (Pikobar) pada Sabtu (11/4/2020) pukul 18.43 WIB, mencatat terdapat 421 orang yang terkonfi
rmasi positif COVID-19(test\jabar11042020.txt)
```

Poin	Ya	Tidak
1. Program berhasil dikompilasi	✓	
2. Program berhasil *running*	✓	
3. Program dapat menerima input dan menuliskan output	✓	
4. Luaran sudah benar untuk data uji		✓

DAFTAR PUSTAKA

http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Pencocokan-String-(2018).pdf

http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf

Modul Praktikum Kuliah Pengantar Regular Expression versi: 18 Feb 2018, update: 11 April 2020 Lisensi: creative commons Yudi Wibisono (yudi@upi.edu), Masayu Leylia Khodra (masayu@informatika.org)