

**IF 2210 STRATEGI ALGORITMA
ALGORITMA PENCOCOKAN STRING
EKSTRAKSI BERITA MENGENAI COVID-19**



Disusun Oleh

Muhammad Ayyub Abdurrahman 13518076

**PRODI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020**

BAB 1

DESKRIPSI SINGKAT

1. Algoritma Knuth-Morris-Pratt(KMP)

Algoritma KMP memanfaatkan informasi yang didapatkan dari partial match sepanjang j karakter saat pencocokan pattern dimulai dari posisi i sehingga kita mengetahui karakter apa saja yang ada di $T[i]..T[i + j - 1]$. Dari informasi ini, terdapat dua kemungkinan untuk menghemat jumlah iterasi yang dilakukan.

Pertama, kita dapat melangkahi beberapa iterasi yang tidak memiliki solusi yang mungkin dengan cara mencoba mencocokkan partial match yang sebelumnya telah ditemukan pada iterasi selanjutnya.

$i=2$: w a w
 $i=3$: w a w o

Pada kasus di atas, dua buah pattern saling berlawanan. Kita mengetahui bahwa pada $i = 2$, $T[3]$ dan $T[4]$ adalah “a” dan “w”, sehingga tidak mungkin “w” dan “a” pada $i = 3$ adalah pattern yang kita cari. Kita dapat melangkahi satu atau lebih iterasi sampai kita menemukan sepasang pattern yang tidak saling berlawanan

$i=2$: w a w
 $i=3$: w a w o

Pada kasus di atas, terdapat sepasang “w” yang beririsan. Kita definisikan bahwa irisan dari dua buah string x dan y adalah urutan karakter terpanjang yang merupakan akhiran dari x dan awalan dari y . Pada kasus ini, irisan dari “waw” dan “wawo” adalah “w”. Secara umum, nilai i yang ingin kita langkahi adalah nilai i yang berkorespondensi terhadap irisan terpanjang dari partial match saat ini.

Selain optimasi di atas (yang merupakan optimasi untuk melangkahi iterasi luar yang tidak perlu), kita juga dapat melakukan optimasi untuk melangkahi iterasi dalam yang tidak perlu.

$i=2$: w a w
 $i=4$: w a w o

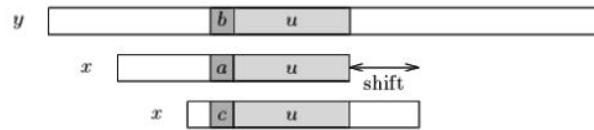
Misal, pada kasus di atas, terdapat pasangan karakter “w” yang beririsan dan diuji oleh iterasi $i = 2$. Seharusnya, kita tidak perlu lagi melakukan pengujian pada $i = 4$. Jika kita sudah memiliki pattern irisan dengan partial match sebelumnya dalam suatu iterasi, kita dapat mengabaikan pengujian sepanjang karakter pattern yang beririsan sebelumnya.

Algoritma KMP ini melakukan paling banyak $2n$ perbandingan, dengan kompleksitas waktu $O(n)$.

2. Algoritma Boyer-Moore

Algoritma Boyer-Moore dinilai sebagai algoritma yang paling efisien. Algoritma ini bekerja dengan memindai karakter-karakter dari pattern yang dimiliki dari ujung paling kanan teks hingga ujung paling kiri, dimulai dari potongan string yang paling kanan dari teks tersebut. Jika string tidak cocok atau string cocok seluruhnya (bukan partial match), algoritma ini akan menggunakan dua buah fungsi untuk menggeser pattern pengujian string matching: good-suffix shift (matching shift) dan bad-character shift (occurrence shift).

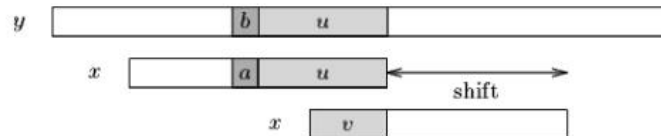
Misal, kita memiliki string x yang merupakan pattern yang ingin kita cari dan string y yang merupakan string teks. Kita asumsikan terjadi ketidakcocokan string pada karakter $x[i] = a$ pada pattern dan karakter $y[i+j] = b$ pada string teks pada sebuah pengujian di posisi j .



Gambar 2.1. good-suffix shift

Sumber : <https://www-igm.univ-mlv.fr/~lecroq/string/node14.html> diakses 22-04-2020 pukul 3.27

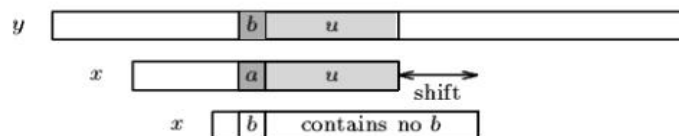
Lalu, $x[i + 1 \dots m - 1] = y[i + j + 1 \dots j + m - 1] = u$ dan $x[i] \neq y[i + j]$. Good-suffix shift menyejajarkan $y[i + j + 1 \dots j + m - 1] = x[i + 1 \dots m - 1]$ dengan partial match pada x yang didahului oleh karakter yang berbeda dari $x[i]$ (lihat gambar di atas). Jika partial match seperti contoh di atas tidak ada, pergeseran yang terjadi adalah pergeseran pattern agar urutan karakter akhir v terpanjang dari $y[i + j + 1 \dots j + m - 1]$ sejajar dengan awalan x yang cocok (lihat gambar 2.2).



Gambar 2.2. Good-suffix shift

Sumber : <https://www-igm.univ-mlv.fr/~lecroq/string/node14.html> diakses 22-04-2020 pukul 3.27

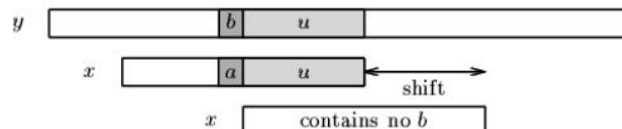
Bad-character shift terdiri dari proses menyejajarkan string teks $y[i + j]$ dengan urutan karakter paling kanan yang cocok dari $x[0 \dots m-2]$, seperti yang dapat dilihat di gambar di bawah ini:



Gambar 2.3. Bad-character shift

Sumber : <https://www-igm.univ-mlv.fr/~lecroq/string/node14.html> diakses 22-04-2020 pukul 3.27

Jika $y[i + j]$ tidak ada pada pattern x , tidak ada urutan karakter pada x yang ada di y dan dapat memenuhi $y[i + j]$, dan ujung kiri dari posisi pengujian disejajarkan dengan $y[i + j + 1]$.



Gambar 2.4. Bad-character shift

Sumber : <https://www-igm.univ-mlv.fr/~lecroq/string/node14.html> diakses 22-04-2020 pukul 3.27

Algoritma BM ini melakukan paling banyak $3n$ perbandingan, dengan kompleksitas waktu $O(n/m)$.

3. Regular Expression

Regular expression (regex) adalah notasi 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 dan bahasa pemrograman sehingga penting untuk dipelajari.

Regular Expression (Regex) secara formal adalah sebuah sekuens dari karakter yang mencerminkan sebuah pola dari string yang akan dicari, tetapi juga bias merujuk pada algoritma yang digunakan untuk mencari pola tersebut. Pola yang ada dalam regex tidak hanya memuat karakter yang memiliki arti secara literal (Contoh : alfabet biasa), tetapi juga memiliki karakter spesial yang dinamakan metacharacter. Metacharacter ini memerintahkan program yang sedang menerima input untuk melakukan sesuatu sesuai dari Metacharacter itu sendiri, dalam shell UNIX metacharacter titik koma “;” memerintahkan program untuk memisahkan pernyataan, sedangkan dalam SQL metacharacter garis bawah “_” memerintahkan program SQL untuk mengganti karakter tersebut menjadi satu huruf lain.

Algoritma regex memiliki Regular language. Regular language adalah sekelompok string yang dapat diekspresikan dengan Regular Expression atau dijadikan input dari Deterministic-finite-automata. Secara formal, Regular Language dari alfabet Σ mengikuti aturan sebagai berikut secara rekursif:

- Bahasa kosong yang dinotasikan sebagai \emptyset , dan Bahasa yang hanya memiliki string kosong yang dinotasikan sebagai $\{\epsilon\}$ adalah Regular Language.
- Untuk setiap a , dimana $a \in \Sigma$ (a merupakan bagian dari Σ), Bahasa singleton $\{a\}$, dimana Bahasa tersebut hanya memiliki is a regular language.
- Jika A dan B adalah Regular Language, maka $A \cup B$ (A digabung dengan B), $A \cdot B$ (A dikonkatenasi dengan B), dan A^* (Kleene star dari A) adalah Regular Language. Sebuah Kleene star dari sebuah sting A adalah penggabungan dari semua pengulangan dari string A : $A^0 \cup A^1 \cup A^2 \cup \dots$
- Tidak ada Bahasa lain dari Σ merupakan Regular Language.

Selain memiliki Regular Language sebuah regex juga harus mengikuti Regular Grammar adalah sebuah objek matematis G , dengan empat komponen, $G = (N, \Sigma, P, S)$, dimana:

- N adalah sebuah himpunan simbol non-terminal, dimana simbol lain dapat diturunkan dari simbol tersebut, yang tidak kosong,
- Σ adalah himpunan simbol terminal, dimana tidak ada simbol lain yang bisa diturunkan dari simbol ini. Nama lain dari himpunan simbol terminal ini adalah alfabet.
- P adalah himpunan dari aturan penurunan dan grammar, dimana setiap aturan memiliki bentuk seperti salah satu dari berikut:
 - $A \rightarrow aB$ (dimana aB diturunkan dari A , dan aB adalah a konkatenasi B)
 - $A \rightarrow a$
 - $A \rightarrow \epsilon$, untuk $A, B \in N$ (A dan B bagian dari N), dan ϵ adalah string kosong.
- $S \in N$ adalah simbol awal

Karena regex memiliki Regular Language maka semua string yang masuk dalam Regular Language termasuk dalam Regular Expression, walau terdapat beberapa perubahan yaitu:

- Union direpresentasikan dengan simbol “|” dan dinamakan sebagai alternasi.
- Urutan dari operasi dari prioritas paling tinggi adalah : Kleene star, konkatenasi, lalu alternasi.

BAB 2

KODE PROGRAM

1. Back-End

a. Utility

```
import os
import nltk
import re
import sys

def getNearestNumbers(Pattern, Kalimat):
    expression = r"(?i)(?:\b)" + Pattern + "(?:\D{0,100})([0-9][0-9,\.]*)(^\.,)|([0-9][0-9,\.]*)(^\.,)(?:\D{0,100})" + Pattern
    listOfAllNumbers = re.findall(expression, Kalimat)
    number = '-1'
    for listOfNumbers in listOfAllNumbers:
        for numbers in listOfNumbers:
            if (numbers != ''):
                number = numbers
    #print(listOfAllNumbers)
    return number

def getAllDates(str):
    expression = r"(?i)(?:\b\D{0,100})((((Kini|Sebelum(nya)?|Hari\s(ini|depan)?|Minggu\s(ini|depan)?|Besok|Lusa|Kemarin|Setelah(nya)?)?((satu|dua|tiga|empat)?(\s)?(se)?hari\s|(se)?pekan\s|(se)?musim))?(\\s)?(Sen(in)?|Sel(asa)?|Rab(u)?|Ka(mis)?|Jumat)?|Sab(tu)?|Min(ggu)?)?(\\s)?(\\s|-|,|/|\\(\\)?(\\s)?((tanggal\\s|tgl\\s)(\\s)(\\d{1,2}))?(\\s)?((\\d{1,2})?(\\s|-|,|\\/)?(\\s)?((\\d{1,2}|Jan(uari)?|Feb(ruari)?|Mar(et)?|Apr(il)?|Mei|Jun(i)?|Jul(i)?|Agu(stus)?|Sep(tember)?|Okt(ober)?|Nov(ember)?|Des(ember)?)(\\s)?(\\s|-|,|\\/)?(\\d{2,4})?)?(\\s)?((tahun\\s)(\\d{2,4}))?)?(\\s)?(\\s|-|,|/|\\(\\)?(\\s)?((pukul)(\\s)?(\\d{2})?(\\s)?(:|\\.)?(\\s)?(\\d{2})?(\\s)?(:)?(\\s)?(\\d{2})?(\\s)?(WIB|WITA|WIT)?)?(\\s)?(pagi|siang|sore|malam)?)(\\s)?(Depan|Lalu|Kemarin)?)"
    array = re.findall(expression, str)
    return array

def getLongestDates(lists):
    max = 0
    kata = "bacod"
    for list in lists:
        for word in list:
            if (len(word) >= max):
                kata = word
                max = len(word)
    return kata

def getRealDates(str):
    listAllDates = getAllDates(str)
    RealDates = getLongestDates(listAllDates)
    listOfDatesFound = re.findall("\\D+", RealDates)
```

```

    if (len(listOfDatesFound) == 1 and listOfDatesFound[0] == ' ') or (len(listOfDatesFound) < 1):
        RealDates = -1
    if RealDates == "bacod":
        RealDates = -1
    return RealDates

```

b. Algoritma KMP

```

import os
import nltk
import re
import sys
import util

def searchWithKMP(Pattern, File):
    #Instansiasi Variabel
    PatternLength = len(Pattern)
    FileLength = len(File)
    cntPattern = 0
    cntFile = 0
    Ada = False

    Prefix = [0]*PatternLength

    borderFunction(Pattern, Prefix)
    while ( cntFile < FileLength ):
        if (Pattern[cntPattern] == File[cntFile] ):
            cntFile += 1
            cntPattern += 1
        if ( cntPattern == PatternLength ):
            Ada = True
            cntPattern = Prefix[cntPattern-1]
        elif cntFile < FileLength and Pattern[cntPattern] != File[cntFile]:
            if cntPattern != 0:
                cntPattern = Prefix[cntPattern-1]
            else:
                cntFile += 1
    return Ada

def borderFunction(Pattern, Prefix):
    # Fungsi yang mengembalikan nilai border Function dari suatu Pattern
    # Masukan : String Pattern
    # Keluaran : List of Integer yang berisi
    PatternLength = len(Pattern)
    cntPrefix = 0
    Prefix[0]
    cntSuffix = 1
    while ( cntSuffix < PatternLength):

```

```

        if (Pattern[cntSuffix] == Pattern[cntPrefix]):
            cntPrefix +=1
            Prefix[cntSuffix] = cntPrefix
            cntSuffix += 1
        else:
            Prefix[cntSuffix] = 0
            cntSuffix += 1

name = sys.argv[1]

fileName = '../test/' #Untuk Ubuntu
fileName2 = '..\\test\\' #Untuk Windows

try:
    fileName += name
    txtFile = open(fileName) #Untuk Ubuntu
except:
    fileName2 += name
    txtFile = open(fileName2) #Untuk Windows
txt = txtFile.read().replace("\n", " ")

pat = sys.argv[2]

ada = False
sent_text = nltk.sent_tokenize(txt)

for sentence in sent_text:
    kalimat = str(sentence)
    ada = searchWithKMP(pat,kalimat)
    if (ada):
        print("Kalimat yang mengandung keyword : ")
        print()
        print(kalimat)
        print()
        print("jumlah : " + str(util.getNearestNumbers(pat,kalimat)))
        waktu = str(util.getRealDates(kalimat))
        if waktu != '-1':
            print("waktu: " + str(util.getRealDates(kalimat)))
        else:
            print("waktu: " + str(sent_text[2]))

```

c. Algoritma BM

```

import os
import nltk
import re
import util
import sys

def searchWithBM(Pattern, File):
    PatternLength = len(Pattern)
    FileLength = len(File)
    Ada = False

    badChar = patternRecognizer(Pattern, PatternLength)

    itrFile = 0
    while(itrFile <= FileLength - PatternLength):
        itrPattern = PatternLength-1
        while itrPattern >= 0 and Pattern[itrPattern] == File[itrFile + itrPattern
]:
            itrPattern -= 1
        if itrPattern < 0:
            Ada = True
            if itrFile + PatternLength < FileLength:
                itrFile += (PatternLength - badChar[ord(File[itrFile + PatternLeng
th]))])
            else:
                itrFile += 1
        else:
            itrFile += max(1, itrPattern - badChar[ord(File[itrFile + itrPattern])
])
    return Ada

def patternRecognizer(Pattern, PatternLength):
    badChar = [-1]*256

    for i in range(PatternLength):
        badChar[ord(Pattern[i])] = i;

    return badChar

name = sys.argv[1]

fileName = '../test/' #Untuk Ubuntu
fileName2 = '..\\test\\' #Untuk Windows

try:
    fileName += name
    txtFile = open(fileName) #Untuk Ubuntu
except:
    fileName2 += name

```



```

    txtFile = open(fileName2) #Untuk Windows
txt = txtFile.read().replace("\n", " ")

pat = sys.argv[2]

ada = False
sent_text = nltk.sent_tokenize(txt)

for sentence in sent_text:
    kalimat = str(sentence)
    ada = searchWithBM(pat,kalimat)
    if (ada):
        print("Kalimat yang mengandung keyword : ")
        print("\n")
        print(kalimat)
        print("\n")
        print("jumlah : " + str(util.getNearestNumbers(pat,kalimat)))
        print("\n")
        waktu = str(util.getRealDates(kalimat))
        if waktu != '-1':
            print("waktu: " + str(util.getRealDates(kalimat)))
        else:
            print("waktu: " + str(sent_text[2]))

```

d. Regex

```

import nltk
import os
import re
import util
import sys

def searchWithRegex(Pattern,File):
    Ada = True
    listOfFound = re.findall(Pattern,File)
    if (len(listOfFound) < 1):
        Ada = False
    return Ada

name = sys.argv[1]

fileName = '../test/' #Untuk Ubuntu
fileName2 = '..\\test\\' #Untuk Windows

try:
    fileName += name
    txtFile = open(fileName) #Untuk Ubuntu
except:
    fileName2 += name

```

```

    txtFile = open(fileName2) #Untuk Windows
txt = txtFile.read().replace("\n", " ")

pat = sys.argv[2]

ada = False
sent_text = nltk.sent_tokenize(txt)

for sentence in sent_text:
    kalimat = str(sentence)
    ada = searchWithRegex(pat,kalimat)
    if (ada):
        print("Kalimat yang mengandung keyword : ")
        print()
        print(kalimat)
        print()
        print("jumlah : " + str(util.getNearestNumbers(pat,kalimat)))
        waktu = str(util.getRealDates(kalimat))
        if waktu != '-1':
            print("waktu: " + str(util.getRealDates(kalimat)))
        else:
            print("waktu: " + str(sent_text[2]))

```

2. Front End

Untuk Front end, karena keterbatasan tempat, saya akan menampilkan 2 kode yang paling penting saja. Untuk kode lebih jelas silahkan lihat di folder src

a. main.php

```

<?php include("a_config.php");?>
<!DOCTYPE HTML>
<html>
<head>
    <?php include("head-tag-contents.php");?>
</head>
<body>

<?php include("design-top.php");?>
<?php include("navigation.php");?>

<div class="container" id="main-content">
    <form method='post' action='hasil.php' enctype='multipart/form-data'>
        <h2>Selamat Datang!</h2>
        <p>Ekstraksi berita dilakukan dengan mencari angka terdekat dari keyword yang anda masukkan dan waktu yang tersedia pada kalimat yang dipilih. Silakan menggunakan website untuk Indonesia yang lebih baik. Silahkan baca petunjuk penggunaan di Tab 'Petunjuk'. Jangan lupa untuk berdoa dan berbuat baik ya!</p>
        <div class="title" id="bagianIsi">
            <br>

```

```

        <h4 style="text-align: left" id= "cmdInputFile"><?php echo 'Masukkan Teks Berita'; ?></h4>
        <input type="file" name="file[]" id="file" multiple>
        <br>
        <br>
        <br>
    </div>
    <h4 style="text-align: left"><?php echo 'Pilih Algoritma'; ?></h4>
    <!--Opsi algoritma yang dapat digunakan-->
    <div class="radio-container">
        <input type="radio" id="KMP" name="algoritma" value="KMP">
        <label class="radio" for="KMP">Knuth-Morris-Pratt</label><br>
        <input type="radio" id="Boyer-Moore" name="algoritma" value="BM">
        <label class="radio" for="Boyer-Moore">Boyer-Moore</label><br>
        <input type="radio" id="Regex" name="algoritma" value="Regex">
        <label class="radio" for="Regex">Regex</label><br>
    </div>
    <br>
    <h4 style="text-align: left">Masukkan Keyword</h4>
    <div class="search-bar">
        <input class="input-box" type="text" placeholder="Masukan keyword" name="keyword" >
        <input type="submit" name="submit" value="submit">
    </div>
    <br>
    <br>
</form>
</div>
<?php include("footer.php");?>

</body>
</html>

```

b. hasil.php

```

<?php include("a_config.php");?>
<!DOCTYPE HTML>
<html>
<head>
    <?php include("head-tag-contents.php");?>
</head>
<body>

<?php include("design-top.php");?>
<?php include("navigation.php");?>

<div class="container" id="main-content">
    <h2>Hasil Ekstraksi</h2>
    <p>Ekstraksi berita dilakukan dengan mencari angka terdekat dari keyword yang anda masukkan dan waktu yang tersedia pada kalimat yang dipilih. Tetap berpikir positif untuk

```

Indonesia yang lebih baik. Hasil ini tidak mempengaruhi masa depan anda! Jangan lupa untuk berdoa dan berbuat baik ya!</p>

```
<div class="title" id="bagianIsi">
<?php
    $keyword = "$_POST[keyword]";
    $algoritma = "<b>$_POST[algoritma]</b>";
    echo "Keyword: $keyword";
    echo "<br> Algoritma yang anda pilih: $algoritma <br>";
    echo "<h5> Hasil Ekstraksi Berita</h5>";
    $countfiles = count($_FILES['file']['name']);
    if(isset($_POST['submit'])){
        // Looping all files
        for($i=0; $i< $countfiles; $i++){
            $filename = $_FILES['file']['name'][$i];

            move_uploaded_file($_FILES['file']['tmp_name'][$i], '..\\test\\'.$filename

e);

            //algoritmanya disini brow
            if ($algoritma == "<b>KMP</b>"){
                echo "<br> File ke-$i : $filename <br>";
                echo shell_exec("python kmp.py $filename $keyword");
            } elseif ($algoritma == "<b>BM</b>") {
                echo "<br> File ke-$i : $filename <br>";
                echo shell_exec("python bm.py $filename $keyword");
            } elseif ($algoritma == "<b>Regex</b>"){
                echo " <br> File ke-$i : $filename <br>";
                echo shell_exec("python rgx.py $filename $keyword");
            } else{
                echo "<br> Algoritma belum dipilih!<br> ";
            }
            echo "<br>";
            // move_uploaded_file($_FILES['file']['tmp_name'][$i], 'upload/'.$filename

e);
        }
    }
?>
<br>
<br>
</div>

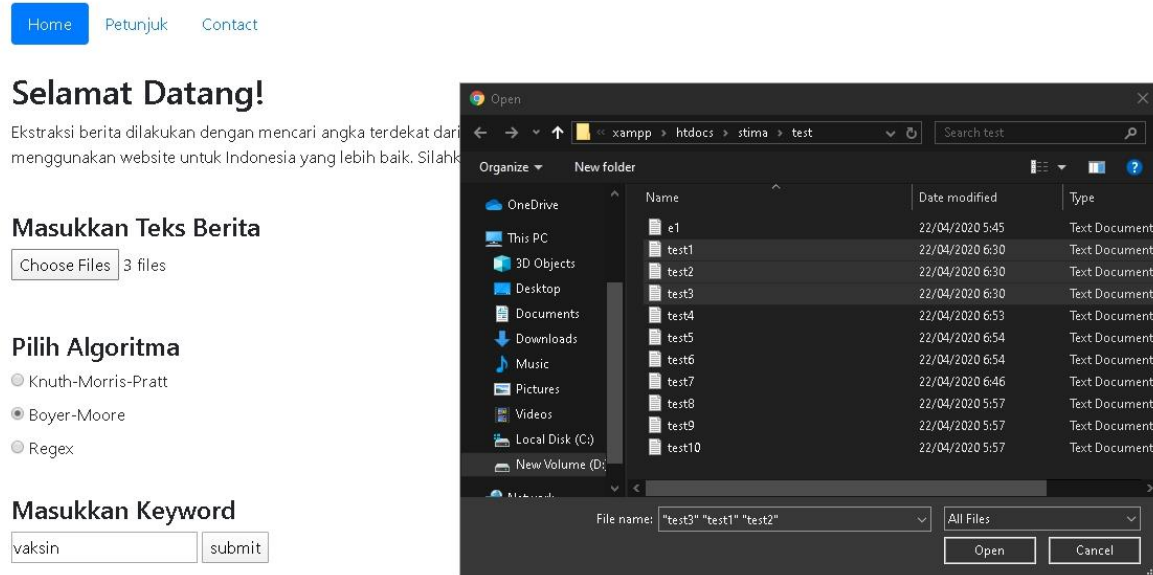
<?php include("footer.php");?>

</body>
</html>
```

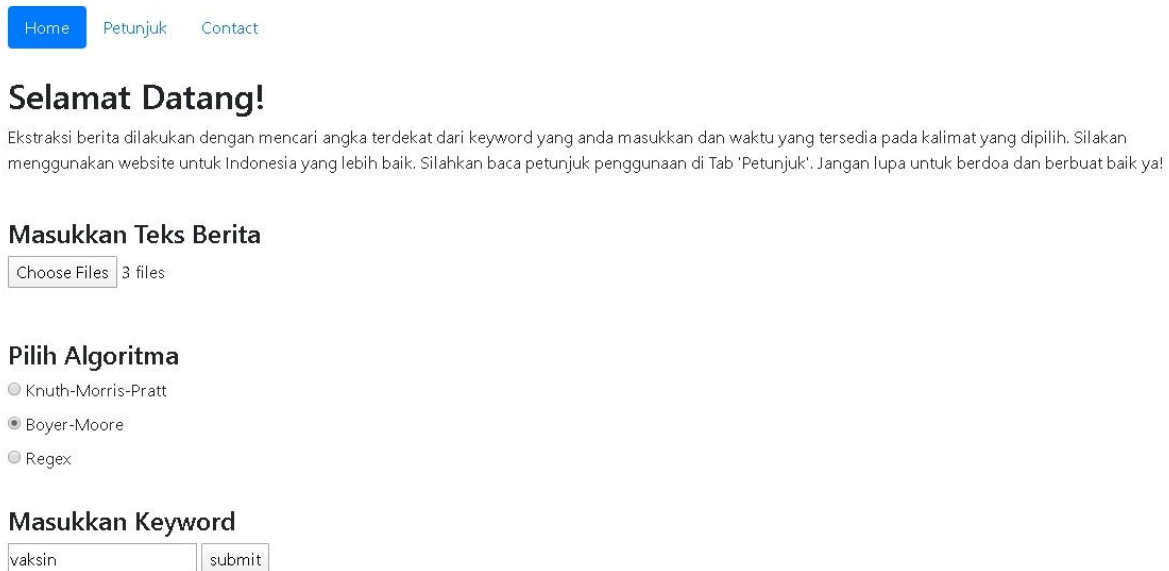
BAB 3

IMPLEMENTASI PROGRAM

1. Test Case untuk Algoritma KMP



Gambar 3.1. Hasil Input untuk Algoritma KMP



Gambar 3.2. Hasil Input untuk Algoritma KMP

Hasil Ekstraksi

Ekstraksi berita dilakukan dengan mencari angka terdekat dari keyword yang anda masukkan dan waktu yang tersedia pada kalimat yang dipilih. Tetap berpikir positif untuk Indonesia yang lebih baik. Hasil ini tidak mempengaruhi masa depan anda! Jangan lupa untuk berdoa dan berbuat baik ya!

Keyword: positif

Algoritma yang anda pilih: **KMP**

Hasil Ekstraksi Berita

File ke-0 : test1.txt

Kalimat yang mengandung keyword : "Pada hari ini kami dapatkan 375 kasus konfirmasi (positif) yang baru, sehingga total ada 7.135 orang," ujar Achmad Yuriyanto, jumlah : 375 waktu: 21/04/2020, 15:50 WIB.

File ke-1 : test2.txt

Kalimat yang mengandung keyword : Bandung - Angka positif virus Corona atau COVID-19 di Jawa Barat menembus angka 400 kasus, jumlah : 19 waktu: Sabtu, 11 Apr 2020 20:07 WIB. Kalimat yang mengandung keyword : Laman Pusat Informasi dan Koordinasi COVID-19 Jabar (Pikobar) pada 11 April 2020 pukul 18.43 WIB, mencatat terdapat 421 orang yang terkonfirmasi positif COVID-19, jumlah : 421 waktu: 11 April 2020 pukul 18.43 WIB Kalimat yang mengandung keyword : Sementara itu, secara nasional terdapat 3.842 kasus positif COVID-19, jumlah : 3.842 waktu: Sabtu, 11 Apr 2020 20:07 WIB.

File ke-2 : test3.txt

Kalimat yang mengandung keyword : Tingkat pemulihan COVID-19 Malaysia terus meningkat 60,74 persen dari total kasus positif, jumlah : 60,74 waktu: 60,74

© Muhammad Ayyub Abdurrahman -- 13518076

Gambar 3.3. Hasil Output untuk Algoritma KMP

2. Test Case untuk Algoritma BM

Selamat Datang!

Ekstraksi berita dilakukan dengan mencari angka terdekat dari keyword yang anda masukkan dan waktu yang tersedia pada kalimat yang dipilih. Silakan menggunakan website untuk Indonesia yang lebih baik. Silahkan baca petunjuk penggunaan di Tab 'Petunjuk'. Jangan lupa untuk berdoa dan berbuat baik ya!

Masukkan Teks Berita

Choose Files 3 files

Pilih Algoritma

- ☐ Knuth-Morris-Pratt
☒ Boyer-Moore
☐ Regex

Masukkan Keyword

Gambar 3.4. Hasil Input untuk Algoritma KMP

Hasil Ekstraksi Berita

File ke-0 : test4.txt

Kalimat yang mengandung keyword : Jakarta - PT Bio Farma (Persero) selaku induk holding badan usaha milik negara (BUMN) sektor farmasi menggandeng China untuk mengembangkan vaksin virus Corona (COVID-19) di dalam negeri. jumlah : 19 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : "Sekarang yang kita lakukan komunikasi dengan lembaga vaksin di China yang namanya Sinovac," kata dia dalam rapat dengar pendapat (RDP) dengan Komisi VI DPR RI secara virtual, Selasa (21/4/2020). jumlah : -1 waktu: Sekarang Kalimat yang mengandung keyword : Dia menjelaskan saat ini China sudah memiliki vaksin yang sedang tahap uji klinis tahap kedua. jumlah : -1 waktu: saat ini Kalimat yang mengandung keyword : Jika vaksin tersebut sukses maka Bio Farma bisa memproduksinya di dalam negeri. jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : "China juga sudah memiliki vaksin yang lagi tahap uji klinis kedua. jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Nah kami juga lagi lakukan koordinasi dengan mereka, bagaimana kalau seandainya vaksin uji klinis berikutnya serta juga untuk proses pembikinan massalnya itu nanti bisa dilakukan di Bio Farma," jelasnya. jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Bio Farma telah mengirimkan proposal ke CEPI untuk ikut serta dalam uji klinis vaksin COVID-19. jumlah : 1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Di samping itu, pihaknya atas inisiatif Kemenristek/BRIN untuk membentuk konsorsium vaksin COVID-19 yang diketuai oleh Lembaga Eijkman dan beranggotakan Bio Farma, Balitbangkes, dan perguruan tinggi. jumlah : 19 waktu: Kamis, 16 Apr 2020 17:30 WIB.

File ke-1 : test5.txt

Kalimat yang mengandung keyword : Jakarta - PT Bio Farma (Persero) berencana melakukan riset dan pengembangan (research and development/R&D) terkait vaksin virus Corona (COVID-19) untuk Indonesia. jumlah : 19 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Pembuatan vaksin ini dilakukan bekerja sama dengan lembaga kesehatan yang didirikan oleh Bill Gates, yakni Koalisi Inovasi Kesiap siagaan Epidemi (The Coalition for Epidemic Preparedness Innovations/CEPI). jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Direktur Utama Bio Farma Honesti Basyir mengatakan lembaga itu sedang dalam tahap pengembangan vaksin di internalnya. jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Kita sudah mengajukan opsi untuk proses scale up (tahap pengujian untuk produksi) untuk vaksin yang lagi dikembangkan. jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Namun, saat ini proses pengembangan vaksin masih dalam skala lab di CEPI, belum masuk ke tahap percobaan klinis. jumlah : -1 waktu: saat ini Kalimat yang mengandung keyword : Namun, masyarakat harus sabar menunggu karena proses

Gambar 3.5. Hasil Output untuk Algoritma Boyer-Moore

File ke-1 : test5.txt

Kalimat yang mengandung keyword : Jakarta - PT Bio Farma (Persero) berencana melakukan riset dan pengembangan (research and development/R&D) terkait vaksin virus Corona (COVID-19) untuk Indonesia. jumlah : 19 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Pembuatan vaksin ini dilakukan bekerja sama dengan lembaga kesehatan yang didirikan oleh Bill Gates, yakni Koalisi Inovasi Kesiap siagaan Epidemi (The Coalition for Epidemic Preparedness Innovations/CEPI). jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Direktur Utama Bio Farma Honesti Basyir mengatakan lembaga itu sedang dalam tahap pengembangan vaksin di internalnya. jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Kita sudah mengajukan opsi untuk proses scale up (tahap pengujian untuk produksi) untuk vaksin yang lagi dikembangkan. jumlah : -1 waktu: Kamis, 16 Apr 2020 17:30 WIB. Kalimat yang mengandung keyword : Namun, saat ini proses pengembangan vaksin masih dalam skala lab di CEPI, belum masuk ke tahap percobaan klinis. jumlah : -1 waktu: saat ini Kalimat yang mengandung keyword : Namun, masyarakat harus sabar menunggu karena proses produksi vaksin ini baru bisa digunakan oleh masyarakat dalam waktu 2-3 tahun mendatang. jumlah : 2 waktu: 2-3 tahun mendatang

File ke-2 : test6.txt

Kalimat yang mengandung keyword : JAKARTA, KOMPAS.com - Holding BUMN Farmasi tengah berupaya menemukan vaksin Covid-19. jumlah : 1 waktu: 21/04/2020, 19:59 WIB. Kalimat yang mengandung keyword : "China juga sudah memiliki vaksin yang lagi tahap uji klinis kedua. jumlah : -1 waktu: 21/04/2020, 19:59 WIB. Kalimat yang mengandung keyword : Nah kami juga lakukan koordinasi dengan mereka, bagaimana kalau seandainya vaksin uji klinis berikutnya, serta untuk proses pembikinan massalnya itu nanti bisa dilakukan di Bio Farma, ujar Honesti saat rapat virtual dengan Komisi VI DPR RI, Selasa (21/4/2020). jumlah : -1 waktu: 21/04/2020, 19:59 WIB. Kalimat yang mengandung keyword : "Lembaga ini dimiliki oleh beberapa negara besar Eropa, mereka sudah sampai penemuan vaksin di lembaga risetnya dan siap melakukan produksi dan uji klinis. jumlah : -1 waktu: 21/04/2020, 19:59 WIB. Kalimat yang mengandung keyword : Dalam hal ini, kita komunikasi dengan CEPI bagaimana scaling up untuk vaksin bisa dilakukan di Bio Farma," kata dia. jumlah : -1 waktu: 21/04/2020, 19:59 WIB. Kalimat yang mengandung keyword : "Target kami 2020 akhir bibit vaksin sudah ada, kemudian 2021 Bio Farma mulai mengembangkan vaksinnya di tempat produksi kami. jumlah : 2021 waktu: 2020 akhir Kalimat yang mengandung keyword : Sehingga target kita di akhir bulan 4 (April) 2021 kita sudah memiliki vaksinnya dan siap dilakukan uji klinis," ucap dia. jumlah : 2021 waktu: 21/04/2020, 19:59 WIB.

Gambar 3.6. Hasil Output untuk Algoritma Boyer-Moore

3. Test Case untuk Regular Expression

The screenshot shows a web application with a navigation bar containing 'Home', 'Petunjuk', and 'Contact'. Below the navigation bar is a large heading 'Selamat Datang!' followed by a paragraph of introductory text. The main section is titled 'Masukkan Teks Berita' and contains a file upload button labeled 'Choose Files' with '4 files' next to it. Below this is a section titled 'Pilih Algoritma' with three radio button options: 'Knuth-Morris-Pratt', 'Boyer-Moore', and 'Regex' (which is selected). At the bottom of this section is a text input field containing the word 'orang' and a 'submit' button.

Home Petunjuk Contact

Selamat Datang!

Ekstraksi berita dilakukan dengan mencari angka terdekat dari keyword yang anda masukkan dan waktu yang tersedia pada kalimat yang dipilih. Silakan menggunakan website untuk Indonesia yang lebih baik. Silahkan baca petunjuk penggunaan di Tab 'Petunjuk'. Jangan lupa untuk berdoa dan berbuat baik ya!

Masukkan Teks Berita

Choose Files 4 files

Pilih Algoritma

☐ Knuth-Morris-Pratt
☐ Boyer-Moore
☒ Regex

Masukkan Keyword

orang submit

Gambar 3.6. Hasil Input untuk Algoritma Regex

Hasil Ekstraksi

Ekstraksi berita dilakukan dengan mencari angka terdekat dari keyword yang anda masukkan dan waktu yang tersedia pada kalimat yang dipilih. Tetap berpikir positif untuk Indonesia yang lebih baik. Hasil ini tidak mempengaruhi masa depan anda! Jangan lupa untuk berdoa dan berbuat baik ya!

Keyword: orang

Algoritma yang anda pilih: **Regex**

Hasil Ekstraksi Berita

File ke-0 : test7.txt

Kalimat yang mengandung keyword : "Salah satu permasalahan sekarang di Indonesia itu kurangnya tes bagi semua masyarakat, sehingga kita bisa mendapatkan informasi yang memadai secara kecukupan data terkait orang yang mungkin potensi mendapat Covid-19," kata Honesti. jumlah : 19, waktu: bacod
Kalimat yang mengandung keyword : "Dengan (kedua alat) ini maka kita harapkan dalam tempo setiap hari ada tes dari 5.000 sampai 10.000 (per hari), maka dalam sebulan kita akan mencapai hampir 300.000 orang yang sudah dites," ujar Arya dalam teleconference dengan wartawan, Rabu (8/4/2020). jumlah : 300.000 waktu: bacod

File ke-1 : test8.txt

Kalimat yang mengandung keyword : Bertambah 72.846 orang dari hari sebelumnya. jumlah : 72.846 waktu: bacod

File ke-2 : test9.txt

Kalimat yang mengandung keyword : "Jadi saya pikir bahwa mulai minggu ini sampai minggu depan per hari akan bisa ada 2.000 hingga 10.000 orang tes per hari sehingga ini modeling kita untuk tahu kapan kira-kira COVID-19 akan berhenti. jumlah : 10.000 waktu: minggu ini

File ke-3 : test10.txt

Gambar 3.6. Hasil Output untuk Algoritma Regex

BAB 4 DATA PENDUKUNG

1. Spesifikasi Komputer

Spesifikasi yang digunakan adalah sebagai berikut

Perangkat Digunakan : Asus A455L

Operating System : Microsoft Windows 10 Pro

Processor : Intel® Core™ i3-5005U CPU @2.00 Ghz (4CPU) ~2Ghz

Memory : 4096 MB RAM

2. Tabel Penilaian

Poin	Ya	Tidak
1. Program berhasil dikompilasi	✓	
2. Program berhasil <i>running</i>	✓	
3. Program dapat menerima input dan menuliskan output.	✓	
4. Luaran sudah benar untuk semua n	✓	