

# **LAPORAN TUGAS KECIL 4**

**Ekstraksi Informasi dari Artikel Berita dengan Algoritma Pencocokan String**

**IF 2211 STRATEGI ALGORITMA**

**KELAS 02**



**Disusun oleh:**

Muhammad Kamal Shafi

13518113

**JURUSAN TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

## DAFTAR ISI

<b>DAFTAR ISI</b>	<b>1</b>
<b>Daftar Gambar</b>	<b>2</b>
<b>BAB I Algoritma Pencocokan String</b>	<b>3</b>
1.1 Algoritma Pencocokan String	3
1.2 Algoritma Knuth-Morris-Pratt	3
1.3 Algoritma Boyer-Moore	4
1.4 Regular Expression pada Pencocokan String	4
<b>BAB II Kode Program</b>	<b>5</b>
2.1.1 Program PatternMatching.py	5
2.1.1.1 Header dan Regular Expression	5
2.1.1.2 Fungsi Algoritma Pencocokan String	6
2.1.1.3 Fungsi untuk Mengekstrak Informasi	8
2.1.2 Program form.py	12
2.1.3 Program InfoExtractionApp.py	13
<b>BAB III Pengujian dan Cek List</b>	<b>16</b>
3.1 Screen-shot Pengujian	16
3.2 Cek List	23

## **Daftar Gambar**

**Gambar 1.** Hasil kasus uji 1

**Gambar 2.** Hasil kasus uji 2

**Gambar 3.** Hasil kasus uji 3

**Gambar 4.** Hasil kasus uji 4

## BAB I Algoritma Pencocokan String

### 1.1 Algoritma Pencocokan String

Algoritma pencocokan string adalah algoritma yang bertujuan untuk menentukan apakah terdapat suatu string(pattern) tertentu didalam suatu string(text) yang lain. Untuk melakukan pencocokan string, terdapat beberapa algoritma/metode umum seperti algoritma Knuth-Morris-Pratt(KMP), Boyer Moore, dan *Regular Expression*.

### 1.2 Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt atau biasa disebut dengan KMP adalah salah satu algoritma pencocokan string yang paling mangkus. Berbeda dengan cara naif, algoritma KMP dapat mencocokkan string dengan kompleksitas waktu  $O(n + m)$  dengan  $n$  dan  $m$  adalah panjang teks dan pattern secara berturut-turut. Algoritma KMP dapat mencapai ini dengan melakukan teknik *precompute*(pemrosesan di awal) pada string pattern. Pemrosesan di awal ini akan menghasilkan suatu array yang menyimpan hasil perhitungan *border function* yang biasanya juga disebut sebagai *prefix function*.

Misalkan terdapat suatu string  $S$  dengan  $|S| = N$ , maka *border function* dari string  $S$  mempunyai satu parameter angka  $i = [0..N - 2]$ . Dari angka  $i$  ini kemudian *border function* akan mengembalikan panjang prefix dari  $S_{0..i}$  dan suffix dari  $S_{1..i}$  yang sama dan bernilai maksimal. Pada string “abcabceda”, *border function* yang dimiliki adalah seperti berikut,

i	0	1	2	3	4	5	6
b(i)	0	0	0	1	2	3	0

Dengan memanfaatkan hasil precompute tersebut, algoritma KMP akan melakukan pencocokan string tanpa mengulangi langkah yang tidak diperlukan kembali.

### 1.3 Algoritma Boyer-Moore

Algoritma Boyer-Moore adalah algoritma pencocokan string yang *heuristic* dengan cara melakukan lompatan karakter yang diamati. Sama seperti KMP, algoritma ini juga melakukan *precompute* yaitu untuk mendapatkan posisi kemunculan terakhir dari suatu karakter yang ada pada string pattern.

Pembandingan karakter dari string pattern dan text pada algoritma ini dilakukan dari karakter paling belakang yang dimiliki oleh string pattern. Terdapat 3 kasus pada saat melakukan pengecekan kecocokan suatu karakter pada string seperti berikut,

misal string pattern adalah  $P$ , string text adalah  $T$  dan pointer  $i$  menunjuk karakter pada string  $T$ , sementara pointer  $j$  menunjuk karakter pada string  $P$ .

1. Jika karakter pada string  $P$  dan  $T$  berbeda, dan setelah dilakukan pengecekan ternyata kemunculan terakhir karakter yang seharusnya, ada di belakang pointer  $j$  saat ini, maka string  $P$  akan digeser sehingga kemunculan terakhir karakter pada string  $P$  berada di posisi yang sama pada pointer  $i$ .
2. Jika karakter pada string  $P$  dan  $T$  berbeda, dan setelah dilakukan pengecekan ternyata kemunculan terakhir karakter yang seharusnya, ada di depan pointer  $j$  saat ini, maka posisi string  $P$  terhadap  $T$  akan geser maju sebesar 1.
3. Jika karakter pada string  $P$  dan  $T$  berbeda, dan ternyata tidak ada karakter yang seharusnya pada string  $P$ , maka string  $P$  akan digeser sebesar  $j + 1$ .

Algoritma Boyer-Moore memiliki kompleksitas waktu  $O(n^2)$  pada kasus terburuknya.

## 1.4 Regular Expression pada Pencocokan String

Regular Expression adalah suatu ekspresi yang mendefinisikan suatu pattern yang dapat diproses oleh mesin pencarian untuk mencari apakah suatu pattern yang telah didefinisikan ini ada pada suatu teks.

## BAB II Kode Program

Pada program terdapat *source code* untuk bagian *front-end* dan *back-end* dari program. Berikut adalah *source code* untuk bagian *back-end* program.

### 2.1.1 Program PatternMatching.py

#### 2.1.1.1 Header dan Regular Expression

```
import os.path
import re
import nltk

# Note:

# abcabcd
# 0 0 0 1 2 3 0 1

# MetaCharacters:
# . ^ $ * + ? { } [ ] \ | ( )

INFINITE = 1000000000

dd = r'(\d{1,2})'
mm = r'(\d{1,2})'
yy = r'(\d{4})'

day = r'(senin|selasa|rabu|kamis|jumat|sabtu|minggu)'
```

```

month =
r'''(januari|februari|maret|april|mei|juni|juli|agustus|september|oktober|novem
ber|desember|
jan|feb|mar|apr|mei|jun|jul|aug|sep|okt|nov|des)'''
tanggal =
r'((({dd}(/|-)({mm}|{month}) (/|-){yy})|({dd}\s{month}\s{yy})).format(dd = dd,
day = day, mm = mm, month = month, yy = yy)
tanggal =
r'({tanggal}|({dd}(/|-)({mm}|{month}))|({dd}\s{month})|({month}\s{yy})).format
(tanggal = tanggal, dd = dd, mm = mm, month = month, yy = yy)

time_zone = r'(WIB|WITA|WIT|UTC)'
hh_mm = r'((([0-1][0-9]|2[0-3])(:|\.)[0-5][0-9]))'
waktu = r'({hh_mm}\s{time_zone}|{hh_mm}).format(hh_mm = hh_mm, time_zone =
time_zone)

integer = r'([1-9]\d*|0)'
biginteger = r'([1-9]\d{0,2}(\.\d{3})*)'
angka = r'({integer},\d+|{biginteger}|{integer}).format(integer = integer,
biginteger = biginteger) # koma = ',' (standard Indonesia)

```

### 2.1.1.2 Fungsi Algoritma Pencocokan String

```

def prefixFunc(s):
    n = len(s)
    pref = []
    pref.append(0)
    le = 0

    for ri in range(1, n):
        while (s[le] != s[ri] and le != 0):
            le = pref[le - 1]
        if (s[le] == s[ri]):
            le += 1
        pref.append(le)

```

```

    return pref

def KMP(s1, s2): # finding pattern s1 in s2
    s1 = s1.lower()
    s2 = s2.lower()
    pref = prefixFunc(s1)
    le = 0
    len1 = len(s1)
    len2 = len(s2)

    result = []
    for ri in range(len2):
        while (s1[le] != s2[ri] and le != 0):
            le = pref[le - 1]
        if (s1[le] == s2[ri]):
            le += 1
        if (le == len1):
            result.append((ri - len1 + 1, ri + 1))
            le = pref[le - 1]
    return result

def boyerMoore(s1, s2): # finding pattern s1 in s2
    s1 = s1.lower()
    s2 = s2.lower()
    len1 = len(s1)
    len2 = len(s2)
    last = [-1] * 256 # ASCII ada 256 character

    result = []
    for i in range(len2):
        last[ord(s2[i])] = i

    p = 0
    while (p + len1 <= len2):
        salah = len1 - 1
        while (salah >= 0):

```



```

        if (s1[salah] == s2[p + salah]):
            salah -= 1
        else:
            break
    if (salah < 0):
        result.append((p, p + len1))
        if (p + len1 >= len2):
            break
        p += max(1, len1 - last[ord(s2[p + len1])])
    else:
        p += max(1, salah - last[ord(s2[p + salah])])
    return result

def regex(s1, s2): # finding pattern s1 in s2
    s1 = s1.lower()
    s2 = s2.lower()
    temp = re.finditer(s1, s2)
    result = []
    for match in temp:
        result.append((match.start(), match.end()))
    return result

```

### 2.1.1.3 Fungsi untuk Mengekstrak Informasi

```

def extractInfo(s1, s2):
    s1 = r' ({s1}) '.format(s1 = s1)
    pat = re.compile(s1, re.I)
    temp = pat.finditer(s2)
    result = []
    for match in temp:
        result.append((match.start(), match.end()))
    return result

def findNearest(x, arr):

```

```

temp = []
for elmt in arr:
    if (x[1] <= elmt[0] + 1):
        temp.append((elmt[0] - x[1] + 1, elmt))
    elif (elmt[1] <= x[0] + 1):
        temp.append((x[0] - elmt[1] + 1, elmt))

    if (len(temp) == 0):
        return None
temp.sort()
return temp[0][1]

def textToCanonical(teks):
    result = []
    sents = nltk.sent_tokenize(teks)
    for sent in sents:
        result.append(nltk.word_tokenize(sent))
    return result

def intersect(pos1, pos2):
    if (pos1[1] < pos2[0]):
        return False
    if (pos1[0] > pos2[1]):
        return False
    return True

def findKeyword(pattern, text, algo): # algo kmp, bm, regex
    sents = nltk.sent_tokenize(text)
    text = textToCanonical(text)
    resOfDay = []
    resOfAlgo = []
    resOfDate = []
    resOfTime = []
    resOfNum = []

    result = []

```

```

hariArtikel = '-'
tanggalArtikel = '-'
jamArtikel = '-'
artIdx = -1

idx = 0
for sent in text:
    temp = ' '
    for word in sent:
        temp = temp + word + ' '
    if (algo == 'kmp'):
        resOfAlgo.append(KMP(pattern, temp))
    elif (algo == 'bm'):
        resOfAlgo.append(boyerMoore(pattern, temp))
    else:
        resOfAlgo.append(regex(pattern, temp))

resOfDay.append(extractInfo(day, temp))
resOfDate.append(extractInfo(tanggal, temp))
resOfTime.append(extractInfo(waktu, temp))
resOfNum.append(extractInfo(angka, temp))

numLen = len(resOfNum[idx])
dateLen = len(resOfDate[idx])
timeLen = len(resOfTime[idx])
# Eliminate intersection
for i in range(numLen):
    for j in range(dateLen):
        if (intersect((resOfNum[idx][i][0] + 1, resOfNum[idx][i][1] -
2), (resOfDate[idx][j][0] + 1, resOfDate[idx][j][1] - 2))):
            resOfNum[idx][i] = (-INFINITE, INFINITE)
            break
    for j in range(timeLen):
        if (intersect((resOfNum[idx][i][0] + 1, resOfNum[idx][i][1] -
2), (resOfTime[idx][j][0] + 1, resOfTime[idx][j][1] - 2))):

```

```

        resOfNum[idx][i] = (-INFINITE, INFINITE)
        break

    for x in resOfAlgo[idx]:
        rangeOfDay = findNearest(x, resOfDay[idx])
        rangeOfDate = findNearest(x, resOfDate[idx])
        rangeOfTime = findNearest(x, resOfTime[idx])
        rangeOfNum = findNearest(x, resOfNum[idx])

        jumlah = '-'
        if (rangeOfNum != None):
            jumlah = temp[rangeOfNum[0] + 1:rangeOfNum[1] - 1]
        waktu_ = '-'
        if (rangeOfDay != None):
            waktu_ = temp[rangeOfDay[0] + 1:rangeOfDay[1] - 1]
        if (rangeOfDate != None):
            if (waktu_ == '-'):
                waktu_ = temp[rangeOfDate[0] + 1:rangeOfDate[1] - 1]
            else:
                waktu_ = waktu_ + ' ' + temp[rangeOfDate[0] +
1:rangeOfDate[1] - 1]
        if (rangeOfTime != None):
            if (waktu_ == '-'):
                waktu_ = temp[rangeOfTime[0] + 1:rangeOfTime[1] - 1]
            else:
                waktu_ = waktu_ + ' ' + temp[rangeOfTime[0] +
1:rangeOfTime[1] - 1]

        result.append(
            {
                'keyword': pattern,
                'jumlah': jumlah,
                'waktu': waktu_,
                'kalimat': sents[idx]
            }
        )

```

```

        if (hariArtikel == '-' and len(resOfDay[idx]) > 0 and artIdx == -1):
            hariArtikel = temp[resOfDay[idx][0][0] + 1: resOfDay[idx][0][1] -
1]

            artIdx = idx

        if (tanggalArtikel == '-' and len(resOfDate[idx]) > 0 and (artIdx == -1
or artIdx == idx)):
            tanggalArtikel = temp[resOfDate[idx][0][0] + 1:
resOfDate[idx][0][1] - 1]
            artIdx = idx

        if (jamArtikel == '-' and len(resOfTime[idx]) > 0 and (artIdx == -1 or
artIdx == idx)):
            jamArtikel = temp[resOfTime[idx][0][0] + 1: resOfTime[idx][0][1] -
1]

            artIdx = idx
            idx += 1

    artikel = '-'
    if (hariArtikel != '-'):
        artikel = hariArtikel
    if (tanggalArtikel != '-'):
        if (artikel == '-'):
            artikel = tanggalArtikel
        else:
            artikel = artikel + ' ' + tanggalArtikel
    if (jamArtikel != '-'):
        if (artikel == '-'):
            artikel = jamArtikel
        else:
            artikel = artikel + ' ' + jamArtikel
    idx = len(result)
    for i in range(idx):
        if (result[i]['waktu'] == '-'):
            result[i]['waktu'] = artikel

    return result

```

### 2.1.2 Program form.py

```
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField, BooleanField
from wtforms.validators import DataRequired, Length, Email, EqualTo

class KeywordForm(FlaskForm):
    keyword = StringField('Keyword', validators = [DataRequired()])
    submit = SubmitField('Extract')
```

### 2.1.3 Program InfoExtractionApp.py

```
import os
import urllib.request
from flask import Flask, render_template, url_for, flash, redirect, request
from forms import KeywordForm
from werkzeug.utils import secure_filename

from PatternMatching import *

UPLOAD_FOLDER = os.path.dirname(__file__) + "\\..\\temp"

app = Flask(__name__)
app.config['SECRET_KEY'] = 'QUNGQANGQINGQONG'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

posts = []

ALLOWED_EXTENSIONS = set(['txt'])

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
```

```

@app.route("/")
@app.route("/home")
def home():
    form = KeywordForm()
    return render_template('home.html', title='App', form = form, posts =
posts)

@app.route("/", methods=['GET', 'POST'])
@app.route("/home", methods=['GET', 'POST'])
def query():
    form = KeywordForm()

    if (request.method == 'POST'):
        print('lagi di home')

        # check if the post request has the file part
        if ('myfile' not in request.files or 'algo' not in request.form):
            flash('No file part / choosen algo', 'danger')
            return redirect(request.url)

        files = request.files.getlist('myfile')
        if (len(files) == 0):
            flash('No file selected for uploading', 'danger')
            return redirect(request.url)

        result = []
        for file in files:
            if (file and allowed_file(file.filename)):
                filename = secure_filename(file.filename)
                teks_path = os.path.join(app.config['UPLOAD_FOLDER'],
"input.txt")
                keyword_path = os.path.join(app.config['UPLOAD_FOLDER'],
"keyword.txt")

                file.save(teks_path)
                kfile = open(keyword_path, 'w')
                kfile.write(form.keyword.data)

```

```

        kfile.close()

        algoPilihan = request.form['algo']
        print(algoPilihan)

        teks = ''
        pattern = ''
        with open(teks_path, 'r') as fileContent:
            teks = fileContent.read()
        with open(keyword_path, 'r') as fileContent:
            pattern = fileContent.read()
        resFile = findKeyword(pattern, teks, algoPilihan)
        resLen = len(resFile)
        for i in range(resLen):
            resFile[i]['filename'] = filename
            result = result + resFile
        else:
            flash('Allowed file types are .txt', 'danger')
            return redirect(request.url)

        posts = result
        flash('extraction successful', 'success')
        return render_template('home.html', title = 'App', form = form, posts =
posts)

        return render_template('home.html', title='App', form = form, posts =
posts)

@app.route("/about")
def about():
    return render_template('about.html', title='About')

if __name__ == '__main__':
    # app.run(debug=True)
    app.run()

```



## BAB III Pengujian dan Cek List

### 3.1 *Screen-shot* Pengujian

1. Input: testcase1.txt

Keyword: terkonfirmasi positif

Algoritma: Knuth Morris Pratt

Isi file:

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.

Hasil:

The screenshot displays the 'Info Extraction Application' interface. At the top, there is a navigation bar with 'Home' and 'About' links. Below the navigation bar, the application is divided into two main sections, each showing the results of a keyword extraction process.

**Section 1:**

- Pilih Algoritma:** Three radio buttons are present: 'Knuth Morris Pratt', 'Boyer Moore', and 'Reguler Expression'. The 'Reguler Expression' option is selected.
- Extract:** A button labeled 'Extract' is visible.
- Keyword: terkonfirmasi positif**
  - Jumlah: 421
  - Waktu: Sabtu 11 Apr 2020 20:07 WIB
- Kalimat:**

" 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. "
- file: testcase1.txt**

**Section 2:**

- Keyword: terkonfirmasi positif**
  - Jumlah: 421
  - Waktu: Sabtu 11/4/2020 18.43 WIB
- Kalimat:**

" 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. "
- file: testcase1.txt**

Gambar 1. Hasil kasus uji 1

2. Input: testcase9.txt

Keyword: ODP

Algoritma: Boyer Moore

Isi file:

Update Corona Jakarta: 3.279 Positif, 878 Tunggu Hasil Tes  
CNN Indonesia | Selasa, 21/04/2020 13:55 WIB

Jakarta, CNN Indonesia -- Jumlah kasus positif terinfeksi virus corona (Covid-19) di DKI Jakarta masih terus bertambah. Hingga Selasa (21/4), secara kumulatif pasien positif corona di Jakarta mencapai 3.279 kasus atau bertambah 167 dari hari sebelumnya.

Angka tersebut diketahui dari laman resmi pemantauan penyebaran virus corona di Jakarta yang diakses CNNIndonesia.com pada pukul 13.30 WIB, Selasa (21/4).

Dari jumlah kasus tersebut, total pasien yang sembuh sebanyak 286 orang. Sementara, jumlah korban meninggal akibat corona hingga hari ini mencapai 305 korban. Masih terdapat 878 orang yang menunggu hasil tes pemeriksaan virus corona.

Sebanyak 1.935 pasien masih menjalani perawatan secara intensif di sejumlah rumah sakit. Sedangkan 753 pasien menjalani isolasi mandiri.

Lebih lanjut, jumlah orang dalam pemantauan (ODP) juga terus bertambah. Per hari ini terdapat 5.799 ODP, dengan rincian, 585 orang masih dalam proses pemantauan dan 5.214 telah selesai dipantau.

Kemudian pasien dalam pengawasan (PDP) hingga saat ini tercatat 5.201 orang, dengan rincian 1.486 masih dirawat, dan 3.715 lainnya dinyatakan sehat dan telah pulang dari perawatan.

Sedangkan, pemakaman jenazah yang dilakukan dengan protap penanganan Covid-19 hingga hari ini di Jakarta sudah mencapai 1.193.

Sebaran pasien positif virus corona paling banyak berada di Kelurahan Petamburan dengan 53 kasus. Kemudian Kelurahan Sunter Agung 45 kasus, Kelurahan Maphar 36 kasus, Kelurahan Kelapa Gadung Barat 33 kasus, Kelurahan Pegadungan 32 kasus, serta Kelurahan Pondok Kelapa 30 kasus.

Virus corona sudah menyebar di 248 kelurahan dari 267 kelurahan yang ada di Jakarta.

Untuk menekan penyebaran virus corona, PSBB akan berakhir pada 23 April mendatang dan dapat diperpanjang jika masih ditemukan penyebaran kasus.

Selama PSBB, Pemprov DKI menyalurkan bantuan sosial kepada warga miskin dan rentan miskin terdampak Covid-19. Penyaluran bantuan ini sudah berlangsung sejak tanggal 9 April dan rencananya akan disalurkan setiap hari hingga 24 April 2020.

Hingga Senin (20/4), Pemprov DKI sudah menyalurkan 840.096 paket bantuan ke 155 kelurahan yang ada di Jakarta. Sementara itu, untuk penyaluran bansos Selasa (21/4), Pemprov akan menyalurkan 92.289 paket bansos ke 18 kelurahan di Jakarta.

Hasil:

ODP

Pilih Algoritma

☐ Knuth Morris Pratt
☐ Boyer Moore
☐ Regular Expression

Extract

Keyword: ODP

- Jumlah: -
- Waktu: Selasa 21/04/2020 13:55 WIB

Kalimat:

"Lebih lanjut, jumlah orang dalam pemantauan (ODP) juga terus bertambah."

file: testcase9.txt

Keyword: ODP

- Jumlah: 5.799
- Waktu: Selasa 21/04/2020 13:55 WIB

Kalimat:

"Per hari ini terdapat 5.799 ODP, dengan rincian, 585 orang masih dalam proses pemantauan dan 5.214 telah selesai dipantau."

file: testcase9.txt

Gambar 2. Hasil kasus uji 2

### 3. Input: testcase10.txt

Keyword: positif

Algoritma: Regular Expression

Isi file:

16 Terindikasi Corona, 95 ABK di Priok Diisolasi di Kapal  
CNN Indonesia | Selasa, 21/04/2020 13:03 WIB

Jakarta, CNN Indonesia -- Polda Metro Jaya bersama Kantor Kesehatan Pelabuhan (KKP) Tanjung Priok melakukan pengecekan risiko infeksi virus corona (Covid-19) terhadap penumpang dan anak buah kapal (ABK) KM Nggapulu yang bersandar di Dermaga 106 Tanjung Priok, Jakarta Utara, Senin (20/4) kemarin.

Kabid Humas Polda Metro Jaya Kombes Yusri Yunus mengatakan kapal dari Surabaya itu membawa sebanyak 161 penumpang dan ABK sebanyak 111 orang.

Pemeriksaan itu, kata Yusri, dilakukan tim kesehatan KKP Tanjung Priok di ruang tunggu Terminal Penumpang Nusantara di Pelabuhan Tanjung Priok pada pukul 13.00-14.00 WIB.

"Hasil pemeriksaan rapid test dari Tim Kesehatan KKP Tanjung Priok ditemukan 16

ABK yang terindikasi Covid-19," kata Yusri dalam keterangannya, Selasa (21/4). Selanjutnya, pada pukul 18.30 WIB, 16 ABK itu dibawa ke Rumah Sakit Peln Petamburan, Jakarta Barat, menggunakan enam mobil ambulans. Sekitar pukul 19.00 WIB, 16 ABK KM Nggapulu itu tiba di RS Peln dan diterima dokter jaga yang saat itu sedang bertugas. Sementara itu, disampaikan Yusri, KM Nggapulu berlabuh di luar dam Pelabuhan Tanjung Priok. Kemudian, untuk 95 ABK lainnya akan melakukan isolasi mandiri di dalam kapal. "95 ABK akan dikarantina mandiri (ODP) di atas kapal selama 14 hari dan mendapat pengawasan dari KKP Tanjung Priok," ucap Yusri. Hingga Senin (20/4) kemarin, pemerintah pusat menyatakan jumlah pasien positif terinfeksi virus corona (Covid-19) di Indonesia sebanyak 6.760 kasus. Dari jumlah itu, sebanyak 590 orang di antaranya meninggal dunia dan 747 orang dinyatakan sembuh.

Hasil:

The screenshot displays the 'Info Extraction Application' interface. At the top, there is a navigation bar with 'Info Extraction Application', 'Home', and 'About'. A green banner at the top of the main content area indicates 'extraction successful'. Below this, the form shows the input details: 'Teks yang ingin di ekstrak' with a 'Browse...' button and 'No files selected.'; 'Keyword' set to 'positif'; and 'Pilih Algoritma' with three radio buttons: 'Knuth Morris Pratt' (selected), 'Boyer Moore', and 'Reguler Expression'. An 'Extract' button is present. The results section shows 'Keyword: positif' with two bullet points: 'Jumlah: 6.760' and 'Waktu: Senin 20/4'. Under 'Kalimat:', a text box contains the sentence: 'Hingga Senin (20/4) kemarin, pemerintah pusat menyatakan jumlah pasien positif terinfeksi virus corona (Covid-19) di Indonesia sebanyak 6.760 kasus.' The file name 'file: testcase10.txt' is shown at the bottom.

Gambar 3. Hasil kasus uji 3

4. Input: testcase2.txt

Keyword: karyawan

Algoritma: Knuth Morris Pratt

Isi file:

Sidak Pabrik saat PSBB, Walkot Bekasi Minta 60 Karyawan Jalani Rapid Test  
Isal Mawardi - detikNews  
Selasa, 21 Apr 2020 02:57 WIB

Bekasi -

Wali Kota Bekasi Rahmat Effendi melakukan sidak ke sejumlah perusahaan yang masih beroperasi saat Pembatasan Sosial Berskala Besar (PSBB). Dua perusahaan itu yakni PT Widya Sakti Kusuma (WSK) dan PT Gold Coin.

Ia meminta 30 karyawan dari masing-masing perusahaan untuk menjalani rapid test. "Petugas Dinas Kesehatan akan saya tugaskan ke 2 PT ini, akan rapid test untuk 30 karyawan yang diambil secara acak di bagian bagian tersebut," ujar Rahmat Effendi dalam keterangannya, Senin (20/4/2020).

Belum diketahui kapan rapid test itu akan dilakukan. Menurut Rahmat, rapid test sangat diperlukan untuk memutus mata rantai penyebaran virus Corona secara cepat.

"Jika salah satu terkena positif (Corona) agar kita langsung lakukan tracking orang orang yang sudah berkomunikasi secara dekat dengan dia," ucap Rahmat.

Kedua perusahaan tersebut, menurut Rahmat, telah melakukan pembatasan operasional. Artinya, sebagian pegawai bekerja dari rumah sementara lainnya tetap bekerja di kantor.

Namun, para karyawan yang bekerja di kantor tetap dituntut menjaga kebersihan dan menjaga jarak (social distancing). Di antaranya seperti menggunakan masker, mencuci tangan, dan wajib disemprot disinfektan bila masuk atau keluar kantor.

Hasil:

Keyword: karyawan

- Jumlah: 30
- Waktu: Selasa 21 Apr 2020 02:57 WIB

Kalimat:

"Ia meminta 30 karyawan dari masing-masing perusahaan untuk menjalani rapid test."

file: testcase2.txt

Keyword: karyawan

- Jumlah: 30
- Waktu: Senin 20/4/2020

Kalimat:

"Petugas Dinas Kesehatan akan saya tugaskan ke 2 PT ini, akan rapid test untuk 30 karyawan yang diambil secara acak di bagian bagian tersebut," ujar Rahmat Effendi dalam keterangannya, Senin (20/4/2020)."

file: testcase2.txt

Keyword: karyawan

- Jumlah: -
- Waktu: Selasa 21 Apr 2020 02:57 WIB

Kalimat:

"Namun, para karyawan yang bekerja di kantor tetap dituntut menjaga kebersihan dan menjaga jarak (social distancing)."

file: testcase2.txt

22

### 3.2 Cek List

Tabel 1. Cek List Program

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 data uji	✓	