Penerapan Algoritma Boyer Moore dalam Pengembangan Fitur Reverse Image Search

Hafshy Yazid Albisthami 13518051
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518051@std.stei.itb.ac.id

Abstrak*—Dalam era digital seperti sekarang ini, *search engine* bukanlah hal yang asing lagi. Dengan semakin banyaknya informasi dan rasa keingintahuan manusia untuk mengetahui banyak hal maka *search engine* atau mesin pencarian merupakan hal yang penting untuk mendapatkan informasi yang akurat sesuai dengan apa yang pengguna cari. Salah satu fitur yang ada pada *search engine* ialah fitur *reverse image search*, dimana pengguna dapat mencari informasi mengenai *file* gambar yang dimiliki. *Reverse image search* sangat bermanfaat bagi banyak kalangan seperti penulis, fotografer, *web master, graphic designer* dan profesi lainnya. Algoritma pencocokan string *Boyer Moore* dapat diimplementasikan untuk mengembangkan fitur *reverse image search* sederhana dengan memanfaatkan algoritma *extract features.

Keywords—search engine; mesin pencarian; reverse image search; algoritma; algoritma Boyer Moore; extract features;

I. PENDAHULUAN

Dalam era digital seperti sekarang ini hampir semua informasi yang berada di dunia mudah untuk dicari. Banyak sekali mesin pencarian yang bisa digunakan secara gratis untuk mencari informasi-informasi yang dibutuhkan. Hanya dengan memasukkan sebuah *input* sederhana, konten-konten dari seluruh bagian internet akan terkumpul. Salah satu mesin pencarian yang paling populer ialah *Google* yang setiap harinya terdapat jutaan orang mengakses layanan pencarian *Google*. Banyak orang yang memanfaatkan *Google* untuk mencari banyak hal dan dengan algoritma yang baik, *Google* dapat mengumpulkan dan mengorganisasikan semua konten dari internet dengan sangat baik.

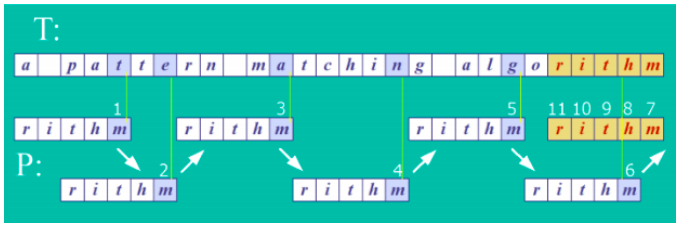
Gambar 1. Fitur *reverse image search* pada mesin pencari *Google*

Salah satu fitur pada *Google* yang paling populer yaitu pencarian dengan menggunakan *input* kata kunci atau *keyword*. Dengan hanya memasukkan sebuah kalimat, mesin pencarian *Google* dapat mencari apa yang pengguna butuhkan. Selain pencarian menggunakan *input* teks kalimat terdapat fitur yang tidak banyak orang menggunakannya namun sangat berguna bagi sebagian orang seperti seseorang yang berprofesi sebagai fotografer, *web master*, *web designer*, *graphic designer* dan pekerjaan lainnya yaitu fitur *reverse image search*.

Reverse image search adalah fitur pencarian berdasarkan *input* sebuah gambar yang memungkinkan penggunanya untuk mendapatkan kesamaan dengan gambar lainnya yang berada di internet sama halnya dengan pencarian menggunakan kalimat, yang membedakannya hanya *input* yang diterima. Mesin pencari *reverse image search* mendapatkan *input* sebuah gambar yang dapat di-*extract* menjadi sebuah larik dua atau tiga dimensi yang berisi angka, lalu mesin tersebut akan mencari gambar yang memiliki kesamaan dengan gambar atau potongan gambar yang berada di internet dan menampilkan hasilnya kepada pengguna. Pencarian gambar tersebut dapat dilakukan dengan menerapkan algoritma pencocokan *pattern* atau pola salah satunya adalah menggunakan algoritma *Boyer Moore*. Dalam makalah ini, akan dibahas secara mendalam cara kerja dan strategi penerapan algoritma *Boyer Moore* dalam pengembangan fitur *reverse image search* sederhana.

II. LANDASAN TEORI

A. *Algoritma Boyer Moore*



Gambar 2. Algoritma Boyer Moore

Pencarian pola atau *pattern* merupakan salah satu hal penting dalam bidang informatika. Saat mencari sebuah *string* dalam suatu teks atau web atau *database*, pencarian pola atau *pattern* menggunakan algoritma dapat digunakan untuk mencari hasilnya. Pada dasarnya algoritma *Boyer Moore* ini

menggunakan dua teknik untuk melakukan pencarian pola atau *pattern* pada sebuah teks yaitu:

1. Teknik *looking-glass*

Teknik *the looking glass* adalah teknik untuk mencari pola P dalam teks T dengan cara bergerak mundur dari P yang dimulai dari akhir P.

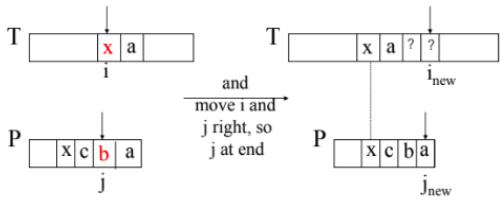
2. Teknik *character-jump*

Teknik *character-jump* dilakukan saat mencapai kondisi seperti:

- Tidak terjadi kecocokan saat teks T ke-i sama dengan x (T[i] == x)
- Karakter pada *pattern* P[j] tidak sama dengan T[i]

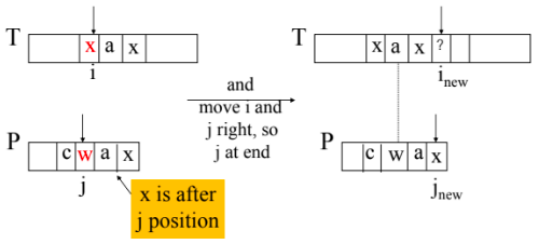
Terdapat tiga kasus yang dapat terjadi secara berurutan yaitu:

1. Jika *pattern* P terdapat x didalamnya, lalu coba lakukan pergeseran P hingga tepat sejajar dengan tempat terakhir dari x di P dengan T[i].



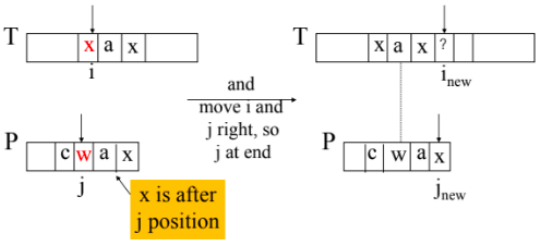
Gambar 3. Kasus 1

2. Jika *pattern* P terdapat x didalamnya, namun menggeser hingga tempat terakhir tidak bisa dilakukan, maka geser P satu karakter hingga T[i+1].



Gambar 4. Kasus 2

3. Jika kasus 1 dan 2 tidak terpenuhi keduanya, maka geser *pattern* P agar P[0] dan T[i+1] sejajar keduanya.



Gambar 5. Kasus 3

Dalam algoritma *Boyer Moore* sebelum melakukan pencarian *pattern* P harus diproses terlebih dahulu dengan membuat *last occurrence function* L(). L() memetakan semua huruf pada *pattern* P menjadi *integer*. L(x) dapat didefinisikan menjadi:

- index i terbesar yang memenuhi P[i] == x, atau
- -1 jika tidak terdapat dimanapun

Contoh *last occurrence function* L():

- P: "strategi"
- A = {a, e, g, i, r, s, t}

P[i]	s	t	r	a	t	e	g	i
i	0	1	2	3	4	5	6	7

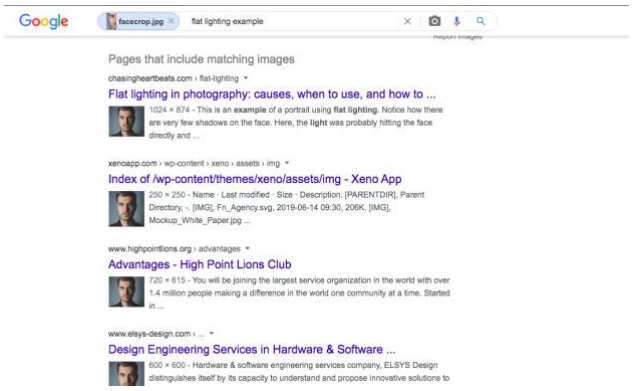
Tabel 1. *Pattern* P

x	a	e	g	i	r	s	t
L(x)	3	5	6	7	2	0	4

Tabel 2. Alfabet A

B. Reverse Image Search

Reverse image search merupakah sebuah fitur sederhana untuk mencari mencari kemiripan sebuah gambar dengan gambar lainnya yang berada di internet. Fitur ini memudahkan penggunanya karena hanya perlu mengunggah gambar yang ingin dicari untuk mencari gambar yang mirip di internet yang sangat luas dan mesin pencari akan mengumpulkan dan mengorgansisaikannya dalam beberapa detik.



Gambar 6. Fitur *reverse image search* pada *Google*

Dengan menggunakan fitur ini pengguna yang tidak mengetahui kata kunci atau *keyword* dapat dengan mudah untuk mencari informasi dengan cara mengunggah contoh gambar yang dimiliki. Selain itu, *rever image search* dapat juga menampilkan semua hasilnya beserta dengan *link* dan artikel yang memuat gambar yang mirip. Beberapa manfaat lainnya dari *reverse image search* yaitu:

1. Mencari tahu lebih dalam suatu objek gambar

Saat pengguna ingin lebih tahu mengetahui objek dari suatu gambar dan tidak dapat mendeskripsikannya, fitur ini dapat mencarikan informasi yang berkaitan dengan gambar yang diunggah.

2. Mencari gambar yang serupa

Terkadang gambar yang pengguna miliki memiliki resolusi yang kurang bagus, dengan menggunakan fitur ini pengguna dapat mencari gambar yang sama namun dengan resolusi yang lebih bagus.

3. Mencari sumber dari suatu gambar

Dalam era digital ini orang bebas untuk mengunduh gambar dan menyebarkannya kembali dengan tujuan yang berbeda. Dengan menggunakan fitur ini gambar dapat dicari sumbernya dan mendapatkan kebenaran dari tujuan gambar tersebut.

4. Menghindari plagiarisme

Saat menuliskan laporan, jurnal, makalah ataupun karya tulis lainnya yang memerlukan pencantuman sumber suatu gambar, fitur ini dapat membantu untuk mencari sumber dari suatu gambar.

5. Mencari tahu produk, tempat, dan seseorang

Untuk mencari informasi lebih lanjut yang lebih bermakna dan bermanfaat seperti harga produk, nama tempat, dan nama seseorang, fitur ini dapat membantu untuk mencarikan semua informasi yang sesuai.

C. Feature Extraction dari data gambar

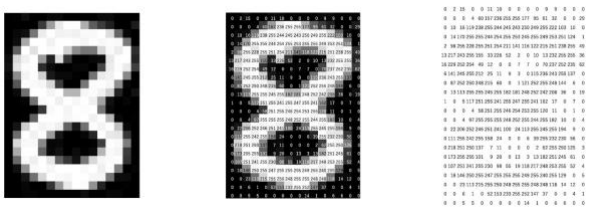
Pada dasarnya sebuah *file* gambar terdiri dari kumpulan persegi-persegi yang berukuran kecil yang memiliki warna solid, yang dapat disebut sebagai *pixel.*

Gambar 7. *File* gambar dengan resolusi kecil

Pada gambar yang memiliki resolusi yang kecil, apabila gambar diperbesar maka akan terlihat seperti gambar (..) yang memiliki beberapa pixel. Komputer menampilkan data gambar seperti gambar (..), dimana pengguna dapat membedakan lekukan dan warna pada suatu gambar. Namun mesin sulit untuk melihat hal tersebut, maka mesin menyimpan data gambar dalam bentuk angka-angka, seperti yang ditampilkan dalam gambar berikut.

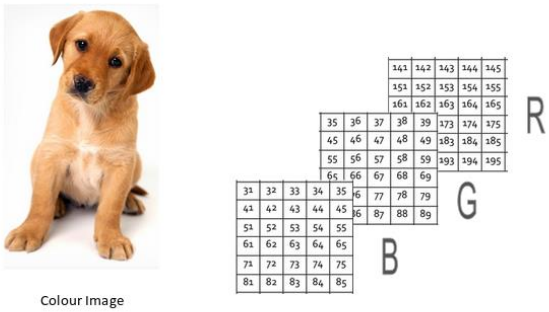
Gambar 8. Representasi mesin menyimpan data gambar

Mesin menyimpan data gambar dalam bentuk matriks kumpulan angka-angka. Besarnya matriks bergantung pada jumlah pixel yang dimiliki gambar. Angka-angka atau nilai *pixel* tersebut merepresentasikan intensitas atau kecerahan untuk tiap *pixel*nya. Semakin nilainya semakin kecil (mendekati angka 0) merepresentasikan warna hitam, sedangkan semakin besar (mendekati angka 255) merepresentasikan warna putih. Jika gambar (..) dan gambar (..) digabung maka akan seperti gambar berikut.



Gambar 9. Representasi data gambar

Itu semua adalah untuk gambar yang hanya memiliki warna hitam dan putih. Pada gambar yang memiliki warna, untuk satu gambar dapat menyimpan tiga buah matriks atau *channels* untuk menyimpan data RGB (*Red, Green, Blue*). Untuk setiap matriks memiliki nilai diantara 0-255 yang merepresentasikan intensitas warna untuk setiap *pixel*.



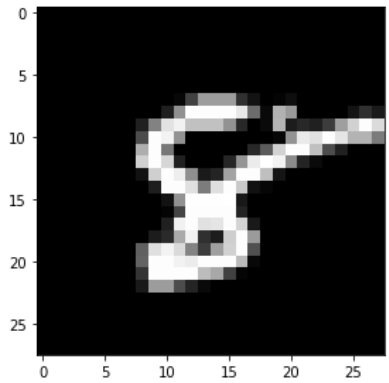
Gambar 10. Representasi data dari gambar berwarna

Manusia dapat melihat gambar anjing. Namun mesin menyimpan data gambar sebagai matriks *Red, Green,* dan *Blue.* Setiap *channel* disatukan dan menghasilkan gambar berwarna.

III. PEMBAHASAN

A. *Membaca data gambar*

Pada pembahasan sebelumnya diterangkan bahwa data gambar direpresentasikan menjadi matriks angka-angka. Sebelum melakukan algoritma *Boyer Moore* untuk menghasilkan kecocokan sebuah gambar, data gambar perlu di-*extract* terlebih dahulu untuk mendapatkan matriks sebuah gambar yang akan menjadi sebuah *pattern* untuk pencocokan menggunakan algoritma *Boyer Moore*. Beberapa bahasa pemrograman memiliki fitur untuk meng-*extract* data gambar, pada pembahasan berikut penulis menggunakan bahasa pemrograman python dan kakas matplotlib, numpy dan skimage untuk melakukannya. Berikut program untuk membaca data gambar.



Gambar 11. image_8_original.png

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from skimage.io import imread, imshow

image = imread('image_8_original.png', as_gray=True)
print("x:" + str(image.shape[0]))
print("y:" + str(image.shape[1]))
print(image)
imshow(image)
plt.show()
```

Algoritma 1. Algoritma pembacaan data gambar

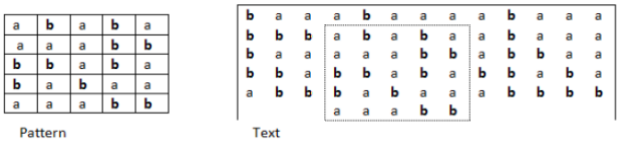
```
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.02745098, 0.25490196, 0.61568627, 0.61568627,
0.61568627, 0.18823529, 0.0745098 , 0.        , 0.01568627,
0.05490196, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
0.12156863, 0.56862745, 0.99215686, 0.99215686, 0.99215686,
0.99215686, 0.90980392, 0.42352941, 0.03529412, 0.69803922,
0.60392157, 0.        , 0.        , 0.        , 0.        ,
0.0745098 , 0.18823529, 0.10196078],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.1254902 , 0.50980392,
0.88627451, 0.94901961, 0.75686275, 0.75686275, 0.75686275,
0.57254902, 0.15686275, 0.        , 0.03921569, 0.70980392,
0.10980392, 0.15686275, 0.1254902 , 0.23529412, 0.62352941,
0.81960784, 0.99215686, 0.85882353],
```

Gambar 12. Output Program

File gambar memiliki dimensi 28x28 yang menghasilkan 784 nilai yang disimpan pada matriks. Terdapat 3 metode yang bisa dipakai untuk melakukan ekstraksi gambar.

B. *Pencocokan pattern menggunakan algoritma Boyer Moore*

Setelah mendapatkan *array* yang didapatkan dari hasil ekstraksi gambar pada pembahasan sebelumnya, selanjutnya adalah mencari kecocokan antara gambar sumber dengan gambar yang berada di internet. Namun pada algoritma *Boyer Moore* biasa, *pattern* yang digunakan adalah *pattern* satu dimensi sedangkan pada pencocokan gambar, hasil yang harus dicocokkan merupakan *array* dua dimensi.



Gambar 13. *Pattern* dan teks dua dimensi

Untuk melakukan hal tersbut maka tidak bisa menggunakan algoritma *Boyer Moore* biasa yang hanya bisa mencocokan *array* satu dimensi, maka diperlukan strategi baru untuk mencocokan *array* dua dimensi tersebut.

1. Membuat *queue* q kosong.
2. Ekstraksi gambar yang ingin dicari menjadi *array pattern* P, dan ekstraksi gambar yang ingin dibandingkan menjadi *array* teks T.
3. Ambil *array* p0 pada *pattern* P berukuran m x n pada *index* ke-0.
4. Bandingkan tiap baris pada teks T berukuran i x j dengan *array* p0 dengan menggunakan algoritma *Boyer Moore* biasa.
5. Apabila ditemukan *pattern* p0 pada teks T, maka masukkan *index* dimana *pattern* ditemukan (x, y) ke dalam *queue* q hingga mencapai *index* i - m.

6. Apabila *queue* kosong, hentikan program dan *pattern* tidak ditemukan.
7. Apabila *queue* tidak kosong, iterasi tiap isi *queue* berupa *index* (x,y) dengan pengecekan algoritma *Boyer Moore* kembali untuk baris setelah pada *pattern* dengan *index* setelah 0 misalkan k, dengan T[x+k].
8. Apabila terjadi kecocokan, maka kembalikan nilai true.
9. Apabila tidak terjadi kecocokan hingga akhir iterasi, maka kembalikan false.

Ilustrasi algoritma *Boyer Moore* pada *array* dua dimensi.

1. q = []

pi	P	
p0	a	b
p1	c	d
p2	a	b

(x, y)	0	1	2	3	4
0	a	c	s	d	a
1	a	d	a	b	a
2	d	d	c	d	d
3	c	c	a	b	c
4	c	c	c	c	c

Tidak terjadi *match*.

2. q = []

pi	P	
p0	a	b
p1	c	d
p2	a	b

(x, y)	0	1	2	3	4
0	a	c	s	d	a
1	a	d	a	b	a
2	d	d	c	d	d
3	c	c	a	b	c
4	c	c	c	c	c

Match pada *index* T[1][2]

3. q = [(1,2)]

pi	P	
p0	a	b
p1	c	d
p2	a	b

(x, y)	0	1	2	3	4
0	a	c	s	d	a
1	a	d	a	b	a
2	d	d	c	d	d
3	c	c	a	b	c
4	c	c	c	c	c

Tidak terjadi *match*, sudah mencapai *index* i - m, lakukan iterasi pada *queue*.

4. q = []

pi	P	
p0	a	b
p1	c	d
p2	a	b

(x, y)	0	1	2	3	4
0	a	c	s	d	a
1	a	d	a	b	a
2	d	d	c	d	d
3	c	c	a	b	c
4	c	c	c	c	c

Match, lanjutkan iterasi.

4. q = []

pi	P	
p0	a	b
p1	c	d
p2	a	b

(x, y)	0	1	2	3	4
0	a	c	s	d	a
1	a	d	a	b	a
2	d	d	c	d	d
3	c	c	a	b	c
4	c	c	c	c	c

Match, sudah mencapai *index* terakhir, kembalikan true.

IV. KESIMPULAN

Algoritma pencocokan *string* seperti *Boyer Moore* dapat digunakan untuk melakukan pencocokan string pada *array* dua dimensi khususnya untuk mengimplementasikan fitur *reverse image search.* Algoritma yang ditulis oleh penulis adalah algoritma yang masih bisa ditingkatkan lagi agar hasil pencarian dapat didapatkan dalam waktu yang lebih cepat dan hasil yang lebih akurat.

TAUTAN VIDEO PADA YOUTUBE

Pengembangan Reverse Image Search menggunakan Algoritma Boyer Moore.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena karunia-Nya telah memberikan penulis kesehatan dan kesempatan untuk menyelesaikan makalah ini di tengah pandemik COVID-19 ini. Penulis juga mengucapkan terima kasih kepada semua pihak yang telah membantu penulis dalam pengerjaan makalah ini baik secara langsung ataupun tidak langsung, khususnya kepada dosen-dosen pengajar mata kuliah IF2211 Strategi Algoritma khususnya Bapak Rinaldi Munir sebagai dosen pengampu kelas 03 atas bimbingannya selama satu semester ini hingga makalah ini dapat diselesaikan.

REFERENSI

[1] 3 Beginner-Friendly Techniques to Extract Features from Image Data using Python, (Online), https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/. Diakses pada 30 April 2020, 18.00 WIB.

[2] Gurram, Sujana. 2011. Image Processing: Pattern Recognition. Indiana State University. Diakses pada 30 April 2020, 18.00 WIB.

[3] Pencocokan String (2018), (Online), http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Pencocokan-String-(2018).pdf. Diakses pada 1 Mei 2020, 20.00 WIB.

[4] Reverse Image Search, (Online), https://www.prepostseo.com/reverse-image-search. Diakses pada 30 April 2020, 20.00 WIB.

[5] Boyer Moore Algorithm for Pattern Searching, (Online), https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/. Diakses pada 30 April 2020, 18.00 WIB.

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, 29 April 2020

Ttd

Nama dan NIM