

# Pengkategorian Tema Puisi Secara Otomatis Menggunakan Pencocokan String

Fatkhan Masruri-13518053

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail: 13518053@std.stei.itb.ac.id

**Abstract**—Puisi merupakan sebuah sastra lama yang sudah tidak asing lagi di Indonesia. Ada banyak banyak judul-judul puisi yang telah diciptakan dengan tema yang berbeda-beda. Pada makalah ini akan dibahas tentang cara pengkategorian puisi berdasarkan tema pada puisi menggunakan metode pencocokan string.

**Kata Kunci**—Puisi, Pencocokan String, Tema.

## I. PENDAHULUAN

Algoritma pencocokan string yang paling umum adalah algoritma *brute force*, cara kerja algoritma ini adalah melakukan penggeseran pada *pattern* di setiap teks, lalu membandingkan setiap karakter pada *pattern*, sampai *pattern* itu ditemukan pada teks atau terjadi ketidakcocokan. Akan karena cara kerja *brute force* kurang maksimal maka banyak algoritma-algoritma pencocokan string yang lain yang kinerjanya lebih baik daripada algoritma *brute force*.

Algoritma pencocokan string dapat diklasifikasikan menjadi tiga berdasarkan arah pencocokan stringnya. Pertama adalah arah yang paling alami, yaitu dari kiri ke kanan. Yang termasuk dalam algoritma ini adalah algoritma *brute force*, algoritma Knuth-Morris-Pratt atau KMP, dan lain sebagainya. Kategori yang kedua adalah kanan ke kiri, kategori ini biasanya menghasilkan hasil terbaik dalam praktiknya. Contoh algoritma yang menerapkan ini adalah algoritma Boyer-Moore, Tuned Boyer Moore, dan Zhu Takaoka. Dan kategori ketiga adalah urutan pembacaan ditentukan secara spesifik oleh algoritma tersebut, arah ini menghasilkan hasil terbaik secara teoritis. Algoritma yang termasuk dalam kategori ini adalah algoritma Colussi, Galil-Seiferas, dan Crochemore-Perrin.

Akibat dari banyaknya algoritma-algoritma yang dikembangkan adalah kesulitan untuk memahami semua algoritma tersebut karena kerja dari setiap algoritma mempunyai ciri khas yang berbeda. Keunikan dari setiap algoritma sangatlah menarik untuk dipelajari dan dipahami. Sebuah algoritma haruslah dilihat untuk dipercaya dan cara yang paling baik untuk memahami sebuah algoritma adalah mencobanya.

Oleh karena itu, pada makalah ini akan memanfaatkan algoritma pencocokan string untuk melakukan pengkategorian

puisi berdasarkan tema secara otomatis. Hal ini dikarenakan puisi merupakan sebuah karya sastra asli Indonesia yang harus dilestarikan.

Banyak puisi karya tokoh terkenal berseliweran di internet. Akan tetapi puisi-puisi itu tidak terkelompokan secara rapi. Sehingga ketika orang ingin mencari referensi puisi dengan tema tertentu akan kesulitan mencarinya. Oleh karena itu, untuk mempermudah pencarian suatu puisi kita perlu menggolongkan puisi-puisi tersebut. Akan tetapi jumlah puisi sekarang sudah lebih dari ribuan dan untuk menggolongkan puisi itu maka perlu usaha yang lebih. Sehingga kita perlu sistem yang dapat melakukan otomatisasi penggolongan puisi tersebut.

## II. DASAR TEORI

### 1. Puisi

Puisi merupakan suatu karya sastra tertulis dimana isinya merupakan ungkapan perasaan seorang penyair dengan menggunakan bahasa yang bermakna semantis serta mengandung irama, rima, dan ritma dalam penyusunan larik dan baitnya. Ada juga yang menyebutkan puisi adalah suatu karya sastra yang isinya mengandung ungkapan kata-kata bermakna, kiasan, dan penyampaiannya disertai dengan rima, irama, larik, dan bait, dengan gaya bahasa yang didapatkan.

Puisi memiliki unsur-unsur yang menyusun sehingga membentuk sebuah puisi yang indah. Unsur-unsur ini adalah sebagai berikut:

#### 1. Tema/Makna

Ini merupakan unsur utama dalam puisi karena dapat menjelaskan makna yang ingin disampaikan oleh seorang penyair dimana medianya berupa bahasa.

#### 2. Rasa (*feeling*)

Rasa adalah sikap sang penyair terhadap suatu masalah yang diungkapkan dalam puisi. Pada Umumnya, ungkapan rasa ini sangat berkaitan dengan latar belakang sang penyair, misalnya agama, pendidikan, kelas sosial, jenis kelamin, pengalaman sosial, dan lain-lain.

### 3. Nada

Nada merupakan sikap seorang penyair terhadap audiensnya serta sangat berkaitan dengan makna dan rasa. Melalui nada, seorang penyair dapat menyampaikan suatu puisi dengan nada mendikte, menggurui, memandang rendah, dan sikap lainnya terhadap audiens.

### 4. Tujuan

Tujuan atau maksud adalah suatu pesan yang ingin disampaikan oleh sang penyair kepada audiensnya.

### 5. Tipografi Puisi

Tipografi merupakan bentuk format suatu puisi, seperti pengaturan baris, tepi kanan-kiri, halaman yang tidak terpenuhi kata-kata. Topografi ini sangat mempengaruhi pemaknaan pada suatu puisi.

### 6. Diksi

Diksi merupakan pemilihan kata yang dilakukan oleh seorang penyair dalam mengungkapkan puisinya sehingga didapatkan efek sesuai dengan yang diinginkan. Pemilihan kata pada puisi sangat berkaitan dengan makna yang ingin disampaikan oleh penyair.

### 7. Gaya Bahasa

Gaya bahasa adalah penggunaan bahasa yang bisa menimbulkan efek dan konotasi tertentu dengan figuratif. Sehingga mengandung banyak makna dalam puisi tersebut. Gaya bahasa sering disebut juga dengan majas. Salah satu contoh majas adalah metafora, ironi, repetisi, pleonasme, dan lain-lain.

### 8. Rima/Irama

Rima/Irama merupakan adanya persamaan bunyi dalam penyampaian puisi, baik di awal, tengah, maupun akhir puisi. Beberapa bentuk rima sebagai berikut:

1. Onotopone, yaitu tiruan terhadap suatu bunyi, misalnya 'ng' yang mengandung efek magis.
2. Bentuk intern pola bunyi, yaitu aliterasi, asonansi, persamaan akhir, persamaan awal, sajak berselang, sajak berpengaruh, sajak penuh, repetisi dan lain sebagainya.
3. Pengulangan kata, yaitu penentuan tinggi-rendah, panjang-pendek, keras-lemah suatu bunyi.

## 2. Tema

Tema merupakan salah satu unsur yang sangat penting dalam sebuah sastra. Tema berkaitan erat dengan fokus ataupun dasar yang dipakai oleh pengarang untuk mengembangkan sebuah cerita atau sastra. Setiap karya

biasanya dibuat dengan berdasarkan tema tertentu dan seluruh aktivitas di dalam cerita juga didasari oleh tema tersebut.

Tema digolongkan menjadi beberapa berikut ini merupakan jenis-jenis tema:

#### 1. Tema Jasmaniah

Tema jasmaniah merupakan tema yang berhubungan atau pun terfokus pada permasalahan kondisi fisik manusia. Model tema ini biasanya menyangkut beberapa hal yang ada di dalam tubuh manusia seperti molekul, jasad, perasaan, tubuh, dan zat. Beberapa contoh tema yang jasmaniah adalah mengenai perasaan cinta.

#### 2. Tema Sosial

Tema sosial merupakan tema yang berkaitan erat dengan berbagai macam hal yang berbau urusan sosial. Dalam tema ini, pengarang cerita biasanya menjelaskan berbagai macam hal yang berkaitan dengan urusan kehidupan masyarakat, interaksi manusia dengan lingkungan sekitarnya, permasalahan sosial, dan berbagai macam tema lainnya.

#### 3. Tema Ketuhanan

Tema Ketuhanan merupakan tema yang berkaitan erat dengan kekuasaan Tuhan yang tampak dalam setiap aktivitas manusia. Model tema ini biasanya dijabarkan oleh pengarang cerita dengan menunjukkan berbagai macam hal – hal magis yang berada di luar akal manusia seperti kejadian kiamat, keajaiban penyembuhan penyakit, dan berbagai macam tema lainnya.

#### 4. Tema Organik

Tema organik merupakan tema yang mencakup berbagai macam hal yang berhubungan erat dengan moral dasar manusia seperti hubungan antar pria dan wanita, nasihat, dan berbagai macam tema lainnya.

#### 5. Tema Egoik

Tema egoik merupakan tema yang berkaitan erat dengan sifat ego manusia. dalam tema ini, pengarang cerita biasanya menonjolkan tema dengan berbagai macam bentuk cerita seperti keserakahan atau pun ketamakan manusia.

### 3. Pencocokan String

Pencocokan string ataupun pencarian string merupakan proses yang melakukan pencarian semua kemunculan string pendek (panjang lebih besar dan lebih kecil daripada teks) yang disebut dengan *pattern* terhadap teks. Untuk melakukan pencocokan string tersebut dapat menggunakan algoritma sebagai berikut:

1. *Brute Force*
2. Knuth-Morris-Pratt
3. Boyer Moore

Selain algoritma di atas juga masih ada algoritma lain yang mungkin lebih baik.

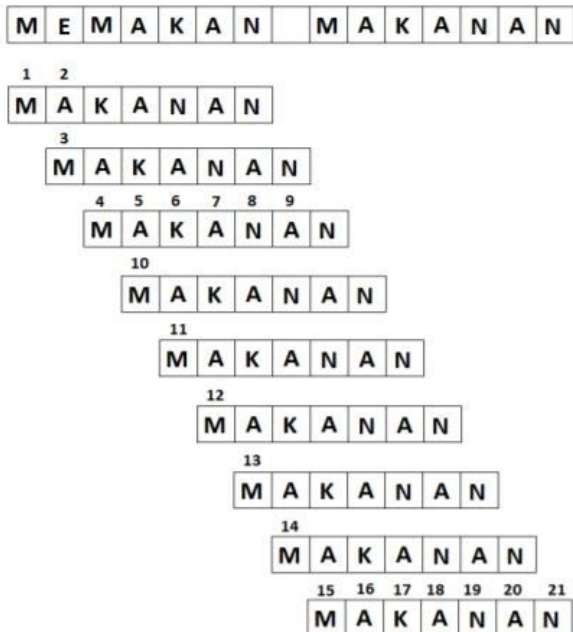
### 3.1 Algoritma Brute Force

Algoritma ini merupakan algoritma yang ditulis dengan pemikiran secara langsung tanpa memikirkan peningkatan performa seperti kompleksitas algoritma. Pseudocode dalam pencocokan string *brute force* adalah berikut ini:

```

procedure BruteForceSearch(
  input m, n : integer,
  input P : array[0..n-1] of char,
  input T : array[0..m-1] of char,
  output ketemu : array[0..m-1] of boolean
)
Deklarasi:
  i, j: integer
Algoritma:
  for (i:=0 to m-n) do
    j:=0
    while (j < n and T[i+j] = P[j]) do
      j:=j+1
    endwhile
    if(j >= n) then
      ketemu[i]:=true;
    endif
  endfor-tema/
  
```

Contoh penyelesaian dengan menggunakan algoritma *brute force*:



Dalam penyelesaian pencocokan string menggunakan *brute force* memiliki *best case*, *worst case*, dan *average case*. Berikut ini merupakan kondisi dimana algoritma *brute force* memiliki kondisi tersebut:

1. Best case
  - Kompleksitas waktu terbaik adalah  $O(n)$
  - Terjadi ketika karakter pertama pada pattern tidak pernah sama dengan karakter yang dicocokkan pada teks
  - Memiliki jumlah perbandingan maksimal sebanyak  $n$  kali
  - Contoh:
 

Teks: isi string ini aah

Pattern:aah
2. Worst case
  - Jumlah perbandingannya adalah  $m(n-m+1)=O(mn)$
  - Contoh:
 

Teks: aaaaaaaaaaah

Pattern:aah
3. Average case
  - Kompleksitas:  $O(m+n)$
  - Contoh:
 

Teks: isi string ini adalah kosong.

Pattern:aah

### 3.2 Algoritma Knuth-Morris-Pratt (KMP)

Pada algoritma *brute force* ketika ditemukan ketidakcocokan pada karakter maka pencocokan akan dimulai dari pattern karakter awal. Sedangkan pada algoritma KMP, informasi yang didapatkan selama pencocokan akan disimpan dan digunakan untuk melakukan pergeseran ketika ditemukan ketidakcocokan. Pergeseran di KMP memanfaatkan fungsi pinggiran atau *border*. Fungsi pinggiran didefinisikan sebagai ukuran awalan terpanjang dari pattern yang merupakan akhiran dari  $P[1...j]$ . Sebagai contoh dari fungsi pinggiran mari kita tinjau pattern  $P = \text{abaaba}$ . Maka nilai pinggirannya adalah sebagai berikut:

$j$	1	2	3	4	5	6
$P[j]$	a	b	a	a	b	a
$b(j)$	0	0	1	1	2	3

Pseudocode untuk mencari nilai pinggiran KMP adalah sebagai berikut ini:

```

procedure preKMP(
  input P : array[0..n-1] of char,
  input n : integer,
  input/output kmpNext : array[0..n] of integer
)
Deklarasi:
i,j: integer

Algoritma
  i := 0;
  j := kmpNext[0] := -1;
  while (i < n) {
    while (j > -1 and not(P[i] = P[j]))
      j := kmpNext[j];
    i:= i+1;
    j:= j+1;
    if (P[i] = P[j])
      kmpNext[i] := kmpNext[j];
    else
      kmpNext[i] := j;
    endif
  endwhile
endwhile

```

Kemudian kita gunakan algoritma untuk mencari kecocokan string pada pattern. Pseudocode untuk algoritma KMP:

```

procedure KMPSearch(
  input m, n : integer,
  input P : array[0..n-1] of char,
  input T : array[0..m-1] of char,
  output ketemu : array[0..m-1] of boolean
)
Deklarasi:
i, j,next: integer
kmpNext : array[0..n] of integer

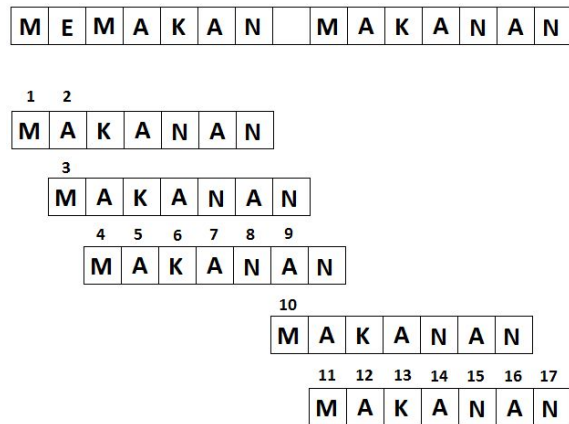
Algoritma:
preKMP(n, P, kmpNext)
i:=0
while (i<= m-n) do
  j:=0
  while (j < n and T[i+j] = P[j]) do
    j:=j+1
  endwhile
  if(j >= n) then
    ketemu[i]:=true;
  endif
  next:= j - kmpNext[j]
  i:= i+next
endwhile

```

Kompleksitas untuk algoritma KMP:

1. Perhitungan fungsi pinggiran:  $O(m)$
2. Pencarian String:  $O(n)$
3. Kompleksitas waktu algoritma KMP adalah  $O(m+n)$

Pencarian yang akan dilakukan oleh algoritma KMP akan seperti pada ilustrasi kasus di bawah ini.



### 3.3 Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan sebuah algoritma pencocokan string yang paling mangkus pada suatu aplikasi pada umumnya. Algoritma ini berbeda dengan algoritma lainnya karena algoritma Boyer-Moore melakukan pencocokan string mulai dari karakter sebelah kanan, bukan dari sebelah kiri. Ide di balik ini adalah bahwa dengan dimulainya pencocokan karakter dari kanan, dan bukan dari kiri maka akan lebih banyak informasi yang didapat. Secara sistematis, langkah-langkah yang dilakukan oleh algoritma Boyer-Moore pada saat mencocokkan string adalah sebagai berikut.

1. Algoritma Boyer-Moore mulai mencocokkan pattern pada awal teks
2. Dari kanan ke kiri, algoritma ini akan memncocokkan karakter per karakter pattern dengan karakter yang ada di teks yang bersesuaian, sampai memenuhi kondisi karakter di pattern dan teks yang dibandingkan tidak cocok atau semua karakter yang ada di pattern cocok.
3. Algoritma kemudian menggeser pattern dengan memaksimalkan nilai pergeseran *good-suffix* dan penggeseran *bad-character*, lalu mengulangi langkah dan sampai pattern yang berada diujung teks.

Pseudocode algoritma Boyer-Moore adalah sebagai berikut:

```

intBMSearch(input m,n:integer
input P:array[0..n-1] of char
input T:array[0...n-1] of char)

Deklarasi
i,j:integer

Algoritma

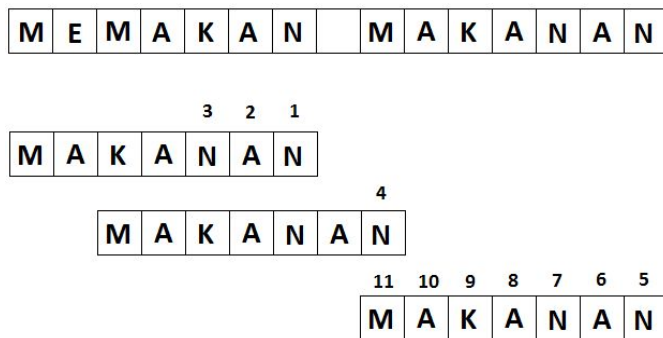
```

```

i <- m-1
j <- m-1
repeat
  it P[j] = T[i] then
    if j = 0 then
      return i {a match!}
    else {cek karakter selanjutnya}
      i <- i-1
      j <- j-1
    else {P[j] <> T[i]}
      j <- i+m-1
      j <- i+max(j-last(T[i],match(i)))
      j <- m-1
until i > n-1
return 0

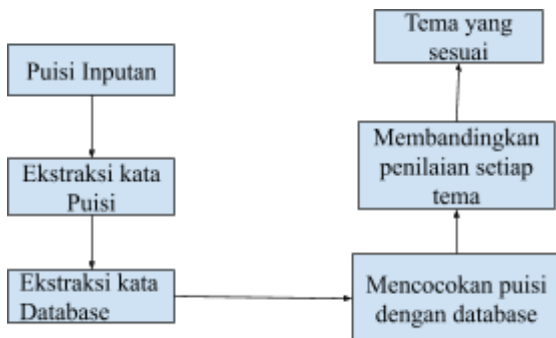
```

Ilustrasi pencocokan string menggunakan algoritma Boyer-Moore



### III. IMPLEMENTASI

Pencarian tema dari sebuah puisi kita memerlukan database untuk menyimpan keyword-keyword yang merujuk ke suatu tema tertentu. Kemudian kita masukan sebuah puisi yang akan kita test puisi tersebut masuk memiliki tema seperti apa. Proses yang akan dilakukan untuk mencari tema dari suatu puisi adalah sebagai berikut.



Berdasarkan bagan di atas tahapan untuk menentukan sebuah tema pada suatu puisi ada 6 tahap. Dan setiap tahap itu

mempunyai sebuah tugas sendiri dan setiap keluaran dari tahap tersebut akan menjadi pertimbangan dalam penentuan hasilnya.

#### 1. Memasukan Puisi

Pada tahap ini adalah kita akan melakukan penginputan puisi yang akan kita lakukan pengecekan temanya. Puisi yang menjadi inputan tidak ada aturan harus berapa baris akan tetapi bahasa yang digunakan oleh puisi tersebut harus disesuaikan dengan bahasa yang digunakan di database. Dalam makalah ini akan digunakan puisi yang berbahasa Indonesia. Contoh puisi

```

Cinta adalah angin
Mengalun lembut ke sela-sela hati
Mencipta sejuk yang sejati
Meniup sedih luka di hati
Hingga hilang tiada kembali

```

#### 2. Ekstraksi kata di puisi

Tahap kedua adalah tahap yang cukup penting yaitu melakukan ekstraksi kalimat inputan menjadi karakter yang dapat diolah oleh program. Tahap ini sangat krusial karena jika hasil ekstraksi berbeda dengan kata atau bentuknya berbeda dengan yang aslinya atau fisiknya maka hasil yang dikeluarkan dapat berbeda.

Ekstraksi kata pada bahasa pemrograman Python dapat menggunakan modul Nltk yang disediakan oleh bahasa pemrograman ini.

```

def openFile(filename):
  text = []
  rslt = []
  tokenizer =
nltk.data.load('tokenizers/punkt/english.pickle')
  fp = open("coba.txt")
  data = fp.read()
  for i in (tokenizer.tokenize(data)):
    text.append(i)
  for i in text:
    for j in i.split(','):
      rslt.append(j)
  return rslt

```

#### 3. Ekstraksi kata database

Database dalam sistem ini berfungsi untuk menyimpan keyword-keyword yang merujuk ke suatu tema. Sehingga setiap tema akan mempunyai database sendiri untuk menyimpan keywordnya. Database ini juga perlu dilakukan ekstraksi untuk mengolah keyword yang ada didalamnya program untuk melakukan ekstraksi database sama dengan

program yang digunakan untuk melakukan ekstraksi pada puisi.

Database ini akan disimpan dalam suatu folder kemudian isi dari folder tersebut file-file dengan nama sesuai dari tema apa yang dirujuk dari keyword yang disimpan dan dengan format text.

Database untuk tema percintaan akan disimpan di file Cinta.txt dengan isi

```
Cinta
Hati
Kita Berdua
```

Database untuk tema alam akan disimpan di file Alam.txt

```
Pegunungan
Lautan
Bumi
Langit
Alam
Debu
Ombak
Angin
```

#### 4. Mencocokkan puisi dengan database

Pada tahap ini kata-kata ada di puisi inputan dan database yang sudah dilakukan ekstraksi kemudian akan dilakukan pengecekan kecocokan. Kata-kata yang ada di database akan dicek apakah puisi tersebut menyimpan kata yang sama dengan database. Kemudian jika puisi tersebut menyimpan salah satu kata yang ada di database maka akan dilakukan vote. Misalnya kata cinta pada suatu puisi ada 5, kemudian kata hati ada 4 maka vote jika puisi itu bertema percintaan adalah 9. Setelah setiap isi dari database percintaan telah dicek kemudian diganti dengan database lain.

Untuk melakukan pengecekan kata pada database kita perlu menggunakan pencocokan string dan algoritma yang akan digunakan telah dijelaskan di bagian sebelumnya.

#### 5. Membandingkan penilaian pada tema

Pada tahap sebelumnya kita telah melakukan pemberian voting pada setiap tema. Kemudian pada tahap ini kita akan melakukan perhitungan voting tersebut untuk menentukan tema mana yang sesuai dengan puisi tersebut. Misalnya pada suatu puisi menghasilkan voting dengan data untuk tema Cinta 9 dan untuk tema Alam 5 maka sistem akan memutuskan bahwa tema dari puisi ini adalah Cinta.

#### 6. Tema yang sesuai

Tahap ini merupakan tahap terakhir yaitu mengeluarkan tema yang sesuai dari puisi yang telah kita masukan. Tahap ini didapat dari keputusan yang telah diambil pada tahap sebelumnya.

### IV. PENGUJIAN

Pengujian menggunakan sistem operasi windows 10 dan compiler yang digunakan adalah Python 3.8.1. Puisi yang digunakan dalam pengujian diambil dari sumber internet yang sudah diketahui temanya. Dalam tahap pengujian hanya digunakan untuk penentuan apakah puisi bertema alam atau bertema cinta. Pengujian menggunakan 2 puisi bertema alam dan 2 puisi bertema cinta. Setiap tema ada puisi yang menggunakan banyak kata kiasan dan ada juga tidak menggunakan bahasa kiasan.

Pengujian ini menggunakan database yang telah dibuat pada bagian sebelumnya.

#### 1. Puisi bertema cinta tanpa kata kiasan

Isi Puisi:

```
Cinta adalah angin
Mengalun lembut ke sela-sela hati
Mencipta sejuk yang sejati
Meniup sedih luka di hati
Hingga hilang tiada kembali
```

Keluaran Program:

```
C:\Users\PS42\Desktop\Kuliah Semester 4\STIMA\tucil4>python algo.py
Cinta adalah angin
Mengalun lembut ke sela-sela hati
Mencipta sejuk yang sejati
Meniup sedih luka di hati
Hingga hilang tiada kembali
Tema Puisi: Cinta
```

Program mengeluarkan keluaran yang sesuai.

#### 2. Puisi bertema cinta dengan penuh kata kiasan

Isi Puisi:

```
Cinta adalah air
Mengalir untuk menyegarkan,
bukan menghanyutkan
apalagi menenggelamkan
```

Keluaran Program:

```
C:\Users\PS42\Desktop\Kuliah Semester 4\STIMA\tucil4>python algo.py
Cinta adalah air

Mengalir untuk menyegarkan,
bukan menghanyutkan
apalagi menenggelamkan

Tema Puisi: Alam
```

Program tidak mengeluarkan hasil yang sesuai.

### 3. Puisi bertema alam tanpa kiasan

Isi Puisi:

```
Awan putih hiasi langit di angkasa
Beraral-arak tanpa saling mendahului
Lintasi gunung sebagai pancangan bumi
Lautan biru terbentang ombak luas
Deburan ombak menyatu dalam biru
```

Keluaran Program:

```
C:\Users\PS42\Desktop\Kuliah Semester 4\STIMA\tucil4>python algo.py
Awan putih hiasi langit di angkasa

Beraral-arak tanpa saling mendahului

Lintasi gunung sebagai pancangan bumi

Lautan biru terbentang ombak luas

Deburan ombak menyatu dalam biru

Tema Puisi: Alam
```

Hasil keluaran program sesuai.

### 4. Puisi bertema alam dengan kiasan

Isi Puisi:

```
Dengan hamparan pasir putihmu
debur ombak yang berdebar
dan keceriaan anak-anak tertawa
tersenyum serta lesung pipimu
bak guratan pasir jemari-jemari lentik
yang sesekali gelombang menyapanya
waktu yang tak pernah kembali
berjalan bahkan berlari
```

Keluaran Program:

```
tersenyum serta lesung pipimu

bak guratan pasir jemari-jemari lentik

yang sesekali gelombang menyapanya

waktu yang tak pernah kembali

berjalan bahkan berlari

Cari Tema Manual
```

Hasil dari program tidak sesuai karena program kata-kata yang ada di database tidak terdapat di puisi.

Dari empat percobaan diatas terlihat program akan berhasil jika isi dari kata puisi tersebut tersurat. Program akan kebingungan dalam menentukan tema ketika isi dari puisi berupa kata kiasan dimana kata yang ditulis tidak sesuai dengan maksudnya.

## V. KESIMPULAN

Sistem ini dapat menentukan tema dari sebuah puisi akan tetapi sistem ini belum bisa dijadikan sebagai penentu dari tema puisi. Hal ini dikarenakan puisi merupakan sebuah karya sastra dimana didalamnya terdapat rima, majas, dan kata kiasan lainnya. Struktur puisi tersebut membentuk sebuah puisi yang indah dan dalam menentukan maksud dari puisi tersebut kita tidak dapat langsung melihat dari isi kata-katanya, karena kebanyakan puisi dibuat dengan makna tersirat. Oleh karena itu dalam penentuan makna atau maksud puisi tersebut harus dibaca dengan sangat mendalam dan memaknai setiap kata-katanya

Akan tetapi, sistem ini dapat digunakan untuk *screening* awal untuk mengelompokan suatu puisi yang acak. Hal ini dikarenakan jika kita melakukan *screening* puisi secara manual dalam jumlah banyak akan memakan banyak waktu. Sehingga sistem ini dapat digunakan untuk pengelompokan awal sehingga waktu yang dibutuhkan tidak selama melakukan *screening* dengan cara manual.

VIDIO LINK DI YOUTUBE

<https://youtu.be/6y0B4DmoVTo>

UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada Tuhan Yang Maha Kuasa karena berkatnya lah penulis dapat menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada semua dosen yang telah membina penulis di mata kuliah IF2210 Strategi Algoritma yaitu Bapak Rinaldi Munir, Ibu Ulfa, dan Ibu Masayu. Selain itu juga penulis mengucapkan terima kasih kepada teman-teman yang sudah mendukung dalam penulisan makalah ini.

## REFERENCES

- [1] Munir, Rinaldi. 2006. Strategi Algoritma. Bandung: Institut Teknologi Bandung.
- [2] Purba, Harry Octavianus. 2014. Algoritma String Matching pada Mesin Pencarian. Makalah.
- [3] "21 Contoh keindahan alam terbaik" - Bangku Depan .<http://bangkudepan.com/contoh-puisi-keindahan-alam/>. Diakses 2020

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.

Kebumen, 2 Mei 2020

A handwritten signature in black ink, appearing to read 'Fatkhan Masruri', written over a horizontal line.

Fatkhan Masruri 13518053