

Aplikasi Algoritma Pencocokkan String Pada Permainan Word Cookies!

Awwala Nisa Kamila - 13519208
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519208@std.stei.itb.ac.id

Abstrak—Word Cookies! adalah salah satu permainan yang menggunakan huruf dalam pelaksanaannya. Permainan ini dapat dimainkan untuk membuat suatu kemungkinan kata baru dari beberapa karakter yang ada. Suatu kemungkinan kata pada permainan ini dapat diselesaikan menggunakan beberapa pendekatan algoritma pencocokkan kata, seperti algoritma brute force, boyer-moore dan algoritma knuth-morris-pratt. Oleh karena itu, pada makalah ini akan dibahas bagaimana sebenarnya cara mencari kata yang sesuai pada permainan Word Cookies! ini.

Kata kunci—Kata, huruf, karakter, permainan

I. PENDAHULUAN

Dewasa ini, hampir 5.22 miliar atau setara dengan 66,6% populasi manusia di bumi merupakan pengguna smartphone dan sebanyak 4.66 miliar diantaranya adalah pengguna internet. Dengan jumlah sebanyak itu, tentunya ada banyak sekali aplikasi yang menarik pada smartphone salah satunya adalah aplikasi permainan. Aplikasi permainan adalah salah satu aplikasi yang banyak dicari oleh pengguna smartphone, baik itu pengguna android ataupun ios.

Genre permainan yang disukai oleh pengguna smartphone salah satunya adalah permainan kata. Hal ini disebabkan permainannya yang cenderung mudah, menarik, dan tentunya dapat membantu mengisi waktu luang. Selain itu, permainan kata juga dapat membantu mengasah otak dan membuat pemain memiliki lebih banyak kosakata. Oleh karena itu, akan dibahas salah satu permainan kata yang cukup menarik pengguna smartphone, yaitu Word Cookies!.

Permainan Word Cookies! adalah suatu permainan yang menampilkan beberapa huruf dan pemain diharuskan untuk membuat suatu kata berarti dari kumpulan huruf tersebut. Permainan ini umumnya menggunakan Bahasa Inggris sehingga dapat membuat pemain menambah kosakata baru dalam Bahasa Inggris. Di sisi lain, permainan ini juga membantu pemain untuk dapat mengetahui ejaan dari suatu kata.

Permainan Word Cookies! menjadi menarik karena semakin tinggi level yang dimiliki, kata yang diharapkan terbentuk menjadi lebih sulit. Hal ini tentunya cukup menantang. Bagi sebagian orang yang tidak terlalu menyukai permainan kata, permainan ini akan terasa sulit dan melelahkan. Namun, bagi penyuka permainan kata tentunya

sangat mengasikkan dan menantang karena dapat menambah kosakata baru yang sebelumnya tidak pernah didengar.

Pada makalah ini, akan terjawab bagaimana menemukan solusi yang efektif untuk menyelesaikan setiap level pada permainan Word Cookies! ini dengan menggunakan beberapa pendekatan algoritma pencocokkan kata, seperti algoritma brute force, Knuth-Morris-Path dan algoritma Boyer-Moore.

II. DASAR TEORI

A. Algoritma Brute Force

Algoritma Brute Force adalah suatu algoritma yang dilakukan dengan cara melakukan pengecekan posisi suatu pola pada suatu teks dimana pencarian dilakukan satu persatu hingga ditemukan pola yang sesuai pada suatu teks.

Salah satu contoh penerapan algoritma Brute Force dengan menggunakan python, yaitu :

```
def bruteforce(pola, teks):  
    '''  
    Berguna untuk mencari pencocokkan string  
    dengan algoritma Brute Force  
    '''  
    PanjangPola = len(pola)  
    PanjangTeks = len(teks)  
  
    j = 0  
    for i in range(0, PanjangTeks - PanjangPola):  
        j = 0  
        while ((j < PanjangPola) and  
(teks[i + j] == pola[j])):  
            j += 1  
            if (j == PanjangPola):  
                return i  
    return -1
```

Kemudian, Langkah-langkah pengerjaan algoritma KMP dapat pula diilustrasikan menjadi sebagai berikut :

Teks: NOBODY NOTICED HIM

Pattern: NOT

```
NOBODY NOTICED HIM
1 NOT
2  NOT
3   NOT
4    NOT
5     NOT
6      NOT
7       NOT
8        NOT
```

Gambar 1 Ilustrasi Langkah pengerjaan algoritma KMP

(Sumber : [3])

Selain Langkah pengerjaan, ada kompleksitas waktu untuk setiap algoritma. Untuk algoritma Brute Force, terdapat 3 jenis kompleksitas waktu yang dimiliki, bila termasuk ke dalam best case, yaitu Ketika pola tidak pernah sama dengan karakter teks yang dicocokkan, kompleksitas waktunya adalah $O(n)$. Selain itu, yang termasuk ke dalam kategori average case adalah Ketika kompleksitas waktunya $O(m+n)$ dan yang termasuk ke dalam worst case Ketika jumlah perbandingannya adalah $m(n-m+1) = O(mn)$

B. Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt atau biasa disebut algoritma KMP adalah suatu algoritma yang dikembangkan secara terpisah oleh 3 orang, yaitu James H. Morris (1966), Vaughan R. Pratt (1966), dan Donald Ervin Knuth (1967). Algoritma ini dilakukan dengan cara pencocokkan string antara suatu teks dan polanya dengan pengurutan dari kiri ke kanan. Cara kerja algoritma KMP mirip seperti pada algoritma brute force tetapi dengan cara mengubah pola menjadi lebih cerdas.

Lebih lengkap, tata cara pencarian dengan algoritma KMP adalah dengan menentukan prefix dan suffix pada pola untuk mencari Panjang dari pola tsb. Kemudian Panjang pola tsb. Digunakan untuk mencari pencocokan string dengan algoritma KMP. Salah satu contoh penerapan algoritma KMP dengan menggunakan python, yaitu :

```
Def KMPSearch(pola, teks):
'''
Berguna untuk mencari pencocokkan string
dengan algoritma KMP
'''
PanjangPola = len(pola)
```

```
PanjangTeks = len(teks)
i = 0
j = 0

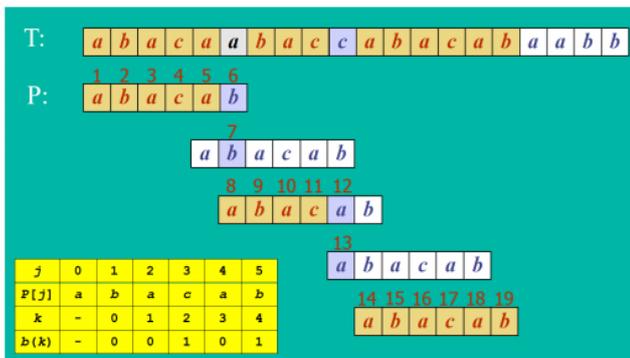
PrefixSuffix = [0 for i in range(PanjangPola)]
hitungPS(pola, PanjangPola, PrefixSuffix)

while (i < PanjangTeks):
    if (pola[j] == teks[i]):
        i += 1
        j += 1
    if (j == PanjangPola):
        print ("Ditemukan Pada : ", i-j)
        j = PrefixSuffix[j-1]
    elif (i < PanjangTeks and pola[j] != teks[i]):
        if (j != 0):
            j = PrefixSuffix[j-1]
        else:
            i += 1
```

```
def hitungPS(pola, PanjangPola, PrefixSuffix):
'''
Befungsi untuk menghitung panjang pola
'''
count = 0
i = 1

while (i < PanjangPola):
    if (pola[i] == pola[count]):
        count += 1
        PrefixSuffix[i] = count
        i += 1
    else:
        if (count != 0):
            count = PrefixSuffix[count-1]
        else:
            PrefixSuffix[i] = 0
            i += 1
```

Kemudian, Langkah-langkah pengerjaan algoritma KMP dapat pula diilustrasikan menjadi sebagai berikut :



Jumlah perbandingan karakter: 19 kali

Gambar 2 Ilustrasi Langkah pengerjaan algoritma KMP
(Sumber : [3])

Selain Langkah pengerjaan, ada kompleksitas waktu untuk setiap algoritma. Untuk algoritma KMP, kompleksitas waktu yang dibutuhkan adalah $O(m+n)$.

C. Algoritma Boyer Moore

Algoritma Boyer Moore atau biasa disebut algoritma BM adalah suatu algoritma yang dilakukan dengan dua Teknik, yaitu teknik *the looking glass* dan teknik *the character jump*. Untuk teknik *the looking glass* sendiri dilakukan dengan cara mencari suatu karakter pada suatu kata dengan cara melakukan pencarian mundur dari karakter terakhirnya. Sedangkan untuk teknik *the character jump* dilakukan dengan cara melakukan lompatan karakter apabila Ketika mencari karakter pada suatu kata ditemukan ketidaksamaan.

Ada beberapa persoalan yang dapat diselesaikan dengan teknik *the character jump*, pertama jika pola mengandung x, maka geser pola ke arah kanan dan sejajarkan dengan posisi kemunculan x terakhir pada pola tsb. Persoalan selanjutnya adalah jika pola mengandung x pada suatu posisi, tetapi Ketika sudah melakukan pergeseran ke kanan tidak bisa, maka geser pola ke kanan sebanya 1 karakter. Terakhir, jika kedua permasalahan diatas tidak ditemukan, maka geser atau lompat pola untuk mensejajarkan pola pertama dengan melewati kata pada posisi sebelumnya.

Secara singkat, tata cara pencarian dengan algoritma BM adalah dengan menjadikan karakternya menjadi bernilai -1. Kemudian dilakukan pencarian untuk menemukan kata yang sesuai dengan kata yang dicari dengan menggunakan algoritma BM. Salah satu contoh penerapan algoritma BM dengan menggunakan python, yaitu :

```

return lastChar

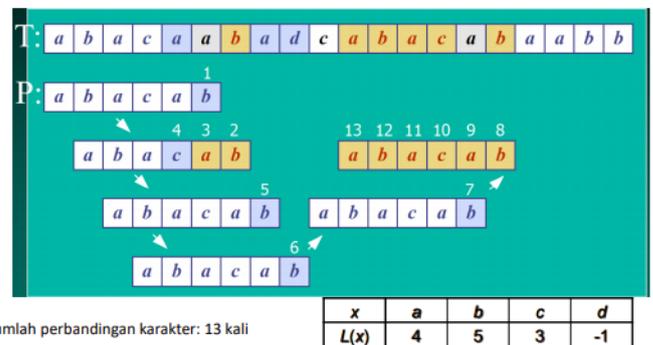
def BM(inputUserArray, command):
'''
Berfungsi untuk menemukan karakter pertama yang
berada pada pola
'''
found = False
for x in command:
    panjangCommand = len(x)
    panjangInputUser = len(inputUserArray)
    lastChar = lastCharHeuristic(x, panjangCommand)
    s = 0
    while(s <= panjangInputUser-panjangCommand):
        j = panjangCommand-1

        while ((j>=0) and (x[j] ==
inputUserArray[s+j])):
            j -= 1

        if (j < 0):
            found = True
            s += (panjangCommand-lastChar[ord
(inputUserArray[s+panjangCommand])])
if s+panjangCommand<panjangInputUser else 1)
        else:
            s += max(1, j-
lastChar[ord(inputUserArray[s+j])])
return found

```

Kemudian, Langkah-langkah pengerjaan algoritma BM dapat pula diilustrasikan menjadi sebagai berikut :



Jumlah perbandingan karakter: 13 kali

Gambar 3 Ilustrasi Langkah pengerjaan algoritma BM
(Sumber : [3])

```

def lastCharHeuristic(string, size):
'''
Berfungsi untuk membuat karakternya bernilai -1
'''
lastChar = [-1]*256
for i in range(size):
    lastChar[ord(string[i])] = i

```

Selain Langkah pengerjaan, ada kompleksitas waktu untuk setiap algoritma. Untuk algoritma BM juga sama dengan KMP, kompleksitas waktu yang dibutuhkannya adalah $O(m+n)$.

III. WORD COOKIES!

Word Cookies! adalah suatu permainan yang menampilkan beberapa huruf dan pemain diharuskan untuk membuat suatu kata berarti dari kumpulan huruf tersebut. Permainan ini umumnya menggunakan Bahasa Inggris sehingga dapat membuat pemain menambah kosakata baru dalam Bahasa Inggris. Di sisi lain, permainan ini juga membantu pemain untuk dapat mengetahui ejaan dari suatu kata. Berikut ini adalah salah satu contoh tampilan dari permainan Word Cookies! :



Gambar 4 Permainan Word Cookies!
(Sumber : Dokumen Pribadi)

IV. ANALISIS APLIKASI ALGORITMA PENCOCOKKAN STRING UNTUK MENYELESAIKAN PERMAINAN WORD COOKIES!

Dalam memecahkan permasalahan pada permainan Word Cookies! dapat digunakan algoritma Knuth-Morris-Pratt dan algoritma Boyer-moore. Pengujian ini dilakukan dengan menggunakan bantuan kamus Bahasa Inggris. Dalam penyelesaiannya akan dilihat hasil pengujian dengan menggunakan bantuan algoritma KMP dan BM.

Langkah-langkah yang dapat dilakukan untuk menyelesaikan permasalahan pada permainan Word Cookies! adalah sebagai berikut :

1. Memasukkan database dari kamus Bahasa Inggris
2. Menerima masukan berupa kemungkinan kata dan Panjang kemungkinan kata yang ingin dicari
3. Membaca data dari database
4. Mencari kemungkinan huruf berdasarkan kata yang disediakan
5. Mencari kata masukan yang sekiranya sesuai atau mirip dengan menggunakan algoritma KMP dan BM
6. Jika kata yang dimasukkan ditemukan pada database, simpan kata dan tampilkan kepada pengguna

A. Algoritma Brute Force

Dengan menggunakan algoritma Brute Force diperoleh hasil analisis untuk pencarian kata untuk menyelesaikan permasalahan pada Gambar 2, yaitu :

```
Masukkan huruf yang tersedia : shwa
Masukkan panjang pola yang ingin dicari : 3
Cara yang ingin digunakan :
1. Algoritma Brute Force
2. Algoritma Knuth-Morris-Pratt
3. Algoritma Boyer-Moore
Masukkan pilihan : 1

Waktu yang dibutuhkan untuk mendapat solusi: 12.7124283829742 s
Kata yang ditemukan :
ash
has
saw
was
```

Gambar 5 Penyelesaian dengan algoritma Brute Force
(Sumber : Dokumen Pribadi)

Dapat dilihat dari gambar di atas bahwa Ketika dimasukkan sebuah input "SHWA" ke dalam persoalan, dan dimasukkan Panjang kata yang ingin dicari, yaitu 3. Kemudian, akan ditemukan kata-kata yang tersusun atas karakter "SHWA" sejumlah Panjang kata yang ingin dicarinya, beserta waktu eksekusi yang dibutuhkan.

B. Algoritma Knuth-Morris-Pratt

Dengan menggunakan algoritma Knuth-Morris-Pratt diperoleh hasil analisis untuk pencarian kata untuk menyelesaikan permasalahan pada Gambar 2, yaitu :

```
Masukkan huruf yang tersedia :shwa
Masukkan panjang pola yang ingin dicari : 3
Cara yang ingin digunakan :
1. Algoritma Brute Force
2. Algoritma Knuth-Morris-Pratt
3. Algoritma Boyer-Moore
Masukkan pilihan : 2

Waktu yang dibutuhkan untuk mendapat solusi: 2.1389849284389 s
Kata yang ditemukan :
ash
has
saw
was
```

Gambar 6 Penyelesaian dengan algoritma KMP

(Sumber : Dokumen Pribadi)

Dapat dilihat dari gambar di atas bahwa Ketika dimasukkan sebuah input “SHWA” ke dalam persoalan, dan dimasukkan Panjang kata yang ingin dicari, yaitu 3. Kemudian, akan ditemukan kata-kata yang tersusun atas karakter “SHWA” sejumlah Panjang kata yang ingin dicarinya, beserta waktu eksekusi yang dibutuhkan.

C. Algoritma Boyer-Moore

Dengan menggunakan algoritma Boyer-Moore diperoleh hasil analisis untuk pencarian kata untuk menyelesaikan permasalahan pada Gambar 2, yaitu :

```
Masukkan huruf yang tersedia :shwa
Masukkan panjang pola yang ingin dicari : 3
Cara yang ingin digunakan :
1. Algoritma Brute Force
2. Algoritma Knuth-Morris-Pratt
3. Algoritma Boyer-Moore
Masukkan pilihan : 3

Waktu yang dibutuhkan untuk mendapat solusi: 0.61274202937429 s
Kata yang ditemukan :
ash
has
saw
was
```

Gambar 6 Penyelesaian dengan algoritma BM

(Sumber : Dokumen Pribadi)

Dapat dilihat dari gambar di atas bahwa Ketika dimasukkan sebuah input “SHWA” ke dalam persoalan, dan dimasukkan Panjang kata yang ingin dicari, yaitu 3. Kemudian, akan ditemukan kata-kata yang tersusun atas karakter “SHWA” sejumlah Panjang kata yang ingin dicarinya, beserta waktu eksekusi yang dibutuhkan.

Berdasarkan hasil uji coba yang telah dilakukan di atas, penggunaan algoritma Brute Force, Knuth-Morris-Pratt, dan algoritma Boyer-Moore dapat dilakukan untuk membantu penyelesaian permainan Word Cookies!. Namun, untuk efektivitas penggunaan, algoritma yang memberikan hasil paling efektif adalah algoritma Boyer-Moore. Selain itu, dari ketiga gambar di atas diketahui juga bahwa pencarian dengan ketiga algoritma menghasilkan jawaban yang sama, tetapi dengan kebutuhan waktu yang berbeda-beda.

V. KESIMPULAN

Dari simulasi yang ditunjukkan di atas, terlepas dari kelebihan dan kekurangannya, dapat diketahui bahwa algoritma pencocokan kata, seperti algoritma brute force, Knuth-Morris-Pratt dan algoritma Boyer-moore dapat digunakan untuk membantu menyelesaikan permainan Word Cookies!. Selain itu, dapat diketahui juga bahwa algoritma yang paling efektif adalah algoritma Boyer-Moore.

VI. LINK VIDEO DI YOUTUBE

Untuk pembahasan lebih jelas mengenai penerapan algoritma pencocokan kata dalam permainan Word Cookies!, penulis menyiapkan video berikut <https://youtu.be/le02vOEHK8A>

VII. UCAPAN TERIMA KASIH

Segala Puji dan syukur penulis panjatkan atas kehadiran Allah Swt. karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan makalah ini tepat waktu. Penulis juga mengucapkan terima kasih kepada orang tua dan kedua adik penulis yang selalu mendukung, serta memberi doa dan semangat kepada penulis. Penulis juga mengucapkan terima kasih kepada ibu/bapak dosen Strategi Algoritma. Terutama Pak Rinaldi selaku dosen mata kuliah Strategi Algoritma untuk K4 yang selama ini selalu memberikan ilmu yang bermanfaat dan juga pemilik website yang berisikan materi kuliah dari tahun ke tahun yang sangat berguna untuk mata kuliah Strategi Algoritma. Penulis juga mengucapkan terima kasih kepada teman-teman penulis yang selalu memberi dukungan dan bantuan.

DAFTAR REFERENSI

- [1] <https://www.liputan6.com/teknoread/4469008/pengguna-internet-dunia-tembus-466-miliar-rata-rata-online-di-smartphone#:~:text=Dengan%205%2C22%20miliar%20orang.sekitar%2093%20juta%20pengguna%20baru> diakses pada Minggu, 9 Mei 2021 pukul 17.53 WIB
- [2] <https://core.ac.uk/display/11730922> diakses pada Minggu, 9 Mei 2021 pukul 18.17 WIB
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf> diakses pada Minggu, 9 Mei 2021 pukul 18.32 WIB
- [4] <https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/> diakses pada Minggu, 9 Mei 2021 pukul 18.32 WIB
- [5] <https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/> diakses pada Minggu, 9 Mei 2021 pukul 18.32 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.

Depok, 26 April 2021

A handwritten signature in black ink that reads "Awwalane". The signature is written in a cursive style with a horizontal line underneath the name.

Awwala Nisa Kamila
13519208