

Aplikasi String Matching dalam Menentukan Asrama Hogwarts yang Sesuai

Flavia Beatrix Leoni A. S. - 13520051

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13520051@std.stei.itb.ac.id

Abstrak—Harry Potter, sebuah serial film Inggris yang diadaptasi dari novel karya J.K. Rowling, menceritakan tentang kisah Harry Potter dan teman-temannya di sekolah sihir Hogwarts. Pada setiap awal tahun ajaran, setiap murid baru akan disortir ke dalam empat asrama, yaitu Gryffindor, Hufflepuff, Ravenclaw, dan Slytherin. Proses penyortiran ini dilakukan oleh topi seleksi berdasarkan kepribadian yang dimiliki. Pada makalah ini, penulis memanfaatkan algoritma *string matching* Boyer Moore dalam melakukan proses penyortiran asrama Hogwarts berdasarkan sifat-sifat yang dimiliki.

Keywords—Harry Potter; Asrama; Topi Seleksi; Kepribadian; Sifat; String Matching; Boyer Moore.

I. PENDAHULUAN

Harry Potter merupakan salah satu seri film fantasi terbaik dan terpopuler yang pernah ada. Film ini diadaptasi dari seri novel karya seorang penulis Inggris, J.K. Rowling yang menceritakan tentang petualangan Harry Potter dan sahabatnya, Ronald Bilius Weasley dan Hermione Jean Granger di Hogwarts, sebuah sekolah khusus yang mempelajari ilmu sihir. Film ini menceritakan upaya Harry dalam mengalahkan Lord Voldemort, penyihir hitam jahat yang berambisi untuk menguasai dunia sihir.

Saat Harry dan siswa lain tiba di Hogwarts untuk pertama kalinya, mereka berkumpul untuk mengikuti perjamuan pemukaan di awal setiap tahun ajaran. Semua siswa berbaris dan nama mereka akan dibacakan sesuai abjad. Kemudian, mereka akan duduk di bangku dan sebuah topi akan diletakkan di atas kepalanya. Topi ini akan berbicara dengan suara pelan kepada pemakainya, menggunakan Legilimency untuk menafsirkan pikiran mereka dan menanggapi. Setelah beberapa saat, topi itu akan mengumumkan asrama yang akan ditempati oleh pemakainya. Terdapat empat asrama di sekolah sihir Hogwarts, yaitu Gryffindor, Hufflepuff, Ravenclaw, dan Slytherin.

Setelah menonton film Harry Potter, banyak orang berandai-andai mengenai asrama Hogwarts mana yang akan mereka tempati. Proses penyortiran oleh topi seleksi dilakukan dengan membaca kepribadian pemakainya dan menyesuaikan dengan kepribadian masing-masing asrama. Oleh karena itu, *string matching* dapat dimanfaatkan untuk melakukan proses penyortiran asrama ini berdasarkan kepribadian yang dimiliki oleh pengguna.

II. LANDASAN TEORI

A. Definisi Pattern/String Matching

Algoritma *string matching* (pencocokan string) adalah algoritma untuk melakukan pencarian semua kemunculan string pendek (*pattern*) yang muncul dalam teks. Terdapat dua komponen dalam *string matching*, yaitu

1. T: teks, yaitu *long string* yang panjangnya n karakter
2. P: *pattern*, yaitu *string* dengan panjang m karakter (asumsi $m \ll n$)

Misalnya, diberikan sebuah teks “the rain in spain stays mainly the plain” dan akan dicari *pattern* “main” dalam teks tersebut, maka *pattern* “main” ditemukan pada index ke-24, yaitu pada “the rain in spain stays **main**ly the plain”.

Terdapat dua konsep mengenai string yang perlu dipahami, yaitu prefix dan suffix. Prefix merupakan huruf atau kelompok huruf yang berada di awal string, sementara suffix merupakan huruf atau kelompok huruf yang berada di bagian akhir string. Misalnya S adalah sebuah string berukuran m , maka prefix dari S adalah substring $S[0..k]$ dan suffix dari S adalah substring $S[k..m-1]$ dimana k adalah sembarang indeks antara 0 dan $m-1$. Contohnya, terdapat string S yaitu “andrew”, maka semua kemungkinan prefix dari S adalah “a”, “an”, “and”, “andr”, “andre”, dan “andrew”, sementara semua kemungkinan suffix dari S adalah “w”, “ew”, “rew”, “drew”, “ndrew”, dan “andrew”.

B. Algoritma Brute Force

Algoritma *brute force* dalam pencocokan string merupakan algoritma yang tidak mempedulikan kinerja atau performa dari suatu sistem. Algoritma ini akan mengecek setiap indeks dari teks mulai dari kiri ke kanan untuk mencari kemunculan *pattern* dalam teks.

Cara kerja dari algoritma ini adalah sebagai berikut.

1. Mula-mula *pattern* dicocokkan pada awal teks.
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap karakter di dalam *pattern* dengan karakter yang bersesuaian di dalam teks hingga semua karakter yang dibandingkan cocok atau sama (pencarian berhasil)

atau dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil).

3. Apabila *pattern* belum ditemukan dalam teks dan teks belum habis, geser *pattern* satu ke kanan dan ulangi langkah 2.

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 Contoh String Matching dengan Algoritma Brute Force
(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)

Pada kasus terburuk, jumlah perbandingan yang harus dilakukan adalah sebanyak $m(n - m + 1) = O(mn)$, contohnya pada kasus

T: aaaaaaaaaaaaaaaaaaaaaah
P: aaah

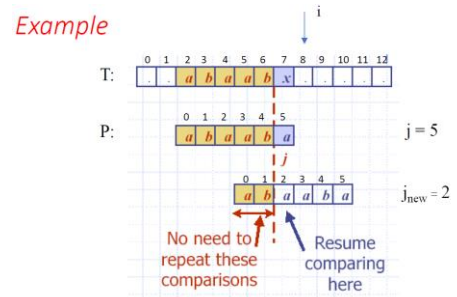
Sementara itu, pada kasus terbaik, kompleksitasnya adalah $O(n)$. Kasus terbaik ini terjadi apabila karakter pertama *pattern* P tidak pernah sama dengan karakter teks T yang dicocokkan dengan jumlah perbandingan maksimal sebanyak n kali. Contohnya adalah

T: String ini berakhir dengan zzz
P: zzz

Algoritma ini akan berjalan dengan cepat ketika terdapat banyak variasi karakter yang digunakan dan akan berjalan dengan lebih lambat apabila variasinya sedikit, misalnya hanya terdapat 0 dan 1. Secara umum, kompleksitas algoritma ini adalah $O(m+n)$.

C. Algoritma Knuth-Morris-Pratt

Algoritma ini mirip seperti algoritma *brute force* yang akan melakukan pencarian *pattern* dalam teks dari kiri ke kanan, tetapi lebih efektif dibandingkan dengan algoritma *brute force*. Pada algoritma ini, apabila terjadi ketidakcocokan antara teks dan *pattern* P di $P[j]$, maka pencarian selanjutnya akan dilakukan dengan melewati prefix terbesar dari $P[0..j-1]$ yang juga merupakan suffix dari $P[1..j-1]$.



Gambar 2 Ilustrasi Algoritma KMP
(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)

Cara kerja dari algoritma Knuth-Morris-Pratt saat mencocokkan string adalah sebagai berikut.

1. Algoritma Knuth-Morris-Pratt mulai mencocokkan *pattern* pada awal teks.
2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang berpadanan hingga karakter di *pattern* dan di teks yang dibandingkan tidak cocok atau seluruh karakter di *pattern* cocok.
3. Apabila *pattern* belum ditemukan dalam teks dan teks belum habis, geser *pattern* berdasarkan tabel next, lalu ulangi langkah 2.

Tabel next merupakan tabel yang dihasilkan dari fungsi pinggiran atau *border function*. Fungsi ini merupakan fungsi yang dapat dilakukan sebagai tahap preprocessing sehingga dapat dilakukan sebelum proses pencarian *pattern* pada teks. Fungsi pinggiran KMP $b(k)$ didefinisikan sebagai jumlah karakter maksimum pada prefix dari *pattern* $P[0..k]$ yang juga merupakan suffix dari *pattern* $P[1..k]$. Misalnya terdapat sebuah *pattern* P “abaaba”, maka nilai fungsi pinggiran dari *pattern* tersebut adalah sebagai berikut.

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

k	0	1	2	3	4
$b(k)$	0	0	1	1	2

Berikut merupakan contoh penerapan algoritma Knuth-Morris-Pratt dalam pencocokan string.



Gambar 3 Contoh String Matching dengan Algoritma Knuth-Morris-Pratt

(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)

Algoritma KMP merupakan modifikasi dari algoritma *brute force* yang lebih cepat dan efisien dengan kompleksitas waktu $O(m+n)$, dengan rincian $O(m)$ untuk menghitung fungsi pinggir dan $O(n)$ untuk pencarian string. Algoritma ini bagus untuk memproses file yang sangat besar karena algoritma ini tidak membutuhkan pergeseran mundur seperti yang dilakukan pada algoritma *brute force*. Namun, algoritma ini kurang dapat bekerja dengan baik apabila variasi alphabetnya semakin banyak karena meningkatkan kemungkinan terjadinya *mismatch* pada awal *pattern* P.

D. Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan algoritma yang dianggap paling efisien dalam melakukan pencocokan string, meskipun langkah yang dilakukan lebih kompleks dibandingkan algoritma-algoritma sebelumnya. Berbeda dengan algoritma sebelumnya, pencocokan string pada algoritma Boyer-Moore dilakukan dari kanan ke kiri.

Algoritma Boyer-Moore merupakan algoritma *pattern matching* yang didasarkan pada dua teknik, yaitu

1. The looking-glass technique

Teknik ini merupakan teknik yang mencari *pattern* P pada teks T dengan bergerak mundur, dimulai dari indeks terakhir.

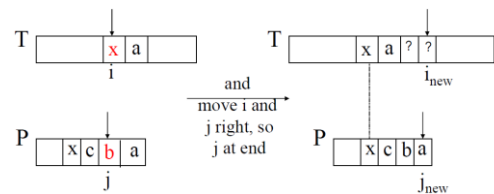
2. The character-jump technique

Apabila *mismatch* terjadi pada $T[i] == x$, dan karakter pada *pattern* $P[j]$ tidak sama dengan $T[i]$, maka dilakukan penggeseran karakter.

Terdapat tiga kemungkinan kasus dalam penggeseran karakter pada teknik *character-jump*, yaitu

1. Kasus pertama

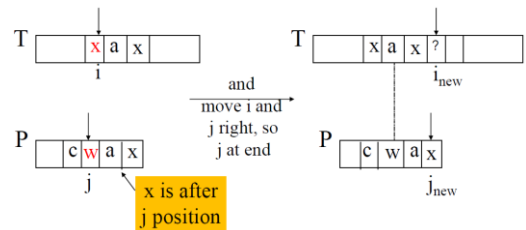
Jika *pattern* P memiliki karakter x di sebelah kiri dari indeks j, maka geser P ke kanan sedemikian rupa sehingga P sejajar dengan indeks i, yaitu indeks dimana x terdekat berada.



Gambar 4 Kasus Pertama Character-Jump Technique
(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)

2. Kasus kedua

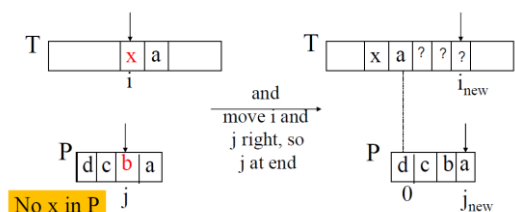
Jika *pattern* P memiliki karakter x, tetapi bukan di sebelah kiri dari indeks j, maka geser P ke kanan sebanyak 1 karakter menuju $T[i+1]$.



Gambar 5 Kasus Kedua Character-Jump Technique
(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)

3. Kasus ketiga

Jika karakter x bukan merupakan substring dari *pattern* P, maka geser *pattern* P sedemikian rupa sehingga *pattern* $P[0]$ sejajar dengan $T[i+1]$

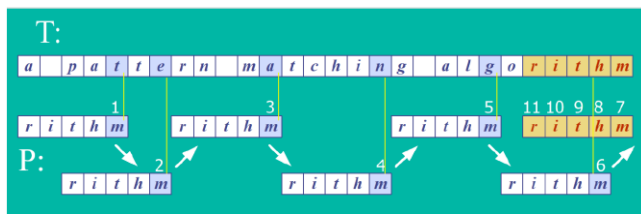


Gambar 6 Kasus Ketiga Character-Jump Technique
(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)

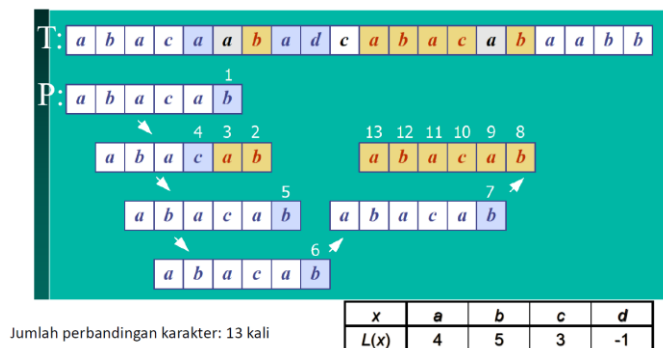
Pada algoritma Boyer-Moore, dibutuhkan fungsi *Last Occurrence* yang memetakan seluruh huruf pada A menjadi sebuah integer yang merepresentasikan posisi terakhir suatu karakter di A. Fungsi *last occurrence* $L(x)$ dengan x merupakan huruf pada A didefinisikan sebagai berikut.

- Indeks I terbesar sedemikian sehingga $P[i] == x$, atau
- -1 jika x tidak terdapat pada A

Berikut merupakan beberapa penerapan algoritma Boyer-Moore dalam pencocokan string.



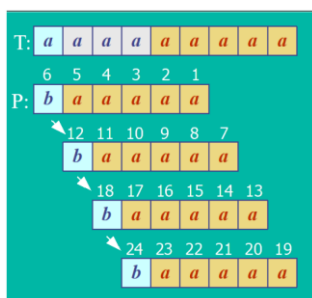
Gambar 7 Contoh 1 String Matching dengan Algoritma Boyer-Moore
(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)



Gambar 8 Contoh 2 String Matching dengan Algoritma Boyer-Moore
(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)

Dalam kasus terburuk, algoritma Boyer-Moore memiliki kompleksitas waktu sebesar $O(nm + A)$ dimana A merupakan jumlah alfabet yang digunakan. Contohnya adalah

T: "aaaaa..a"
P: "baaaaa"



Gambar 9 Contoh Kasus Terburuk String Matching dengan Algoritma Boyer-Moore
(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>, diakses pada 8 Mei 2022)

Algoritma ini cocok digunakan apabila variasi karakter yang digunakan luas atau jumlah alfabet A banyak, misalnya pada English text. Sementara itu, algoritma ini akan berjalan lebih lambat jika jumlah alfabet A kecil, misalnya pada binary. Selain itu, algoritma ini lebih cepat dibandingkan dengan algoritma *brute force*.

III. ASRAMA HOGWARTS

Asrama Hogwarts adalah asrama yang digunakan para murid di sekolah sihir Hogwarts sebagai tempat tinggal selama belajar di Hogwarts. Pembagian asrama ini didasarkan pada seleksi yang dilakukan oleh *sorting hat* atau topi seleksi. Pada seri Harry Potter, sekolah sihir Hogwarts dibagi menjadi empat asrama, dimana masing-masing asrama tersebut menggunakan nama keluarga pendirinya, yaitu Godric Gryffindor, Salazar Slytherin, Rowena Ravenclaw, dan Helga Hufflepuff. Asrama-asrama tersebut saling bersaing selama mereka bersekolah dengan cara menambah atau mengurangi angka dari berbagai peristiwa untuk memperoleh Piala Asrama. Masing-masing asrama memiliki tim Quidditch tersendiri yang bersaing dalam memperebutkan Piala Quidditch.

A. Gryffindor

Gryffindor merupakan salah satu dari empat asrama yang ada di sekolah sihir Hogwarts. Nama Gryffindor diambil dari salah satu pendiri sekolah tersebut, yaitu Godric Gryffindor yang merupakan salah satu penyihir terbesar di masanya. Binatang identitas dari asrama ini adalah singa. Merah dan emas adalah warna kebanggaan Gryffindor. Murid-murid yang tinggal di asrama ini dikenal sebagai orang yang gagah berani, bersifat ksatria, dan tidak mengenal takut, meskipun terkadang sampai ke titik agresivitas dan kecerobohan.

B. Hufflepuff

Hufflepuff didirikan oleh Helga Hufflepuff yang mengutamakan kerja keras, kesabaran, kesetiaan, dedikasi, toleransi, dan keadilan di atas semuanya. Binatang identitas dari asrama ini adalah badger dan kenari dengan kuning dan hitam sebagai warna kebanggaannya. The Fat Friar adalah hantu penjaga asrama ini. Hufflepuff bersesuaian dengan elemen tanah, dimana asrama tempat tinggal mereka berlokasi di ruangan bawah tanah, tepatnya di dekat dapur Hogwarts.

C. Ravenclaw

Ravenclaw merupakan asrama lain yang terdapat di sekolah sihir Hogwarts yang menilai tinggi kecerdasan, kreativitas, akal, kebijaksanaan, dan individualistis. Ravenclaw ditemukan oleh Rowena Ravenclaw. Lambang dari asrama ini adalah elang dan menggunakan warna biru dan perunggu sebagai warna identitas. Hantu asrama ini adalah Grey Lady. Ravenclaw berhubungan dengan elemen udara.

D. Slytherin

Salazar Slytherin merupakan pendiri asrama ini. Asrama ini mengutamakan ambisi, kecerdikan/licik, bakat kepemimpinan, daya akal yang panjang, kecenderungan melanggar aturan dan keturunan darah murni. Selain itu, haus kekuasaan juga merupakan karakteristik dari Slytherin. Binatang yang melambangkan Slytherin adalah ular dan menggunakan warna hijau dan perak sebagai warna identitas. Bloody Baron adalah hantu asrama Slytherin. Slytherin berhubungan dengan elemen air.

IV. APLIKASI STRING MATCHING DALAM MENENTUKAN ASRAMA HOGWARTS YANG SESUAI BERDASARKAN SIFAT YANG DIMILIKI

Masing-masing asrama yang ada di Hogwarts memiliki sifat-sifat yang menggambarkan murid yang tinggal di dalamnya. Oleh karena itu, dapat ditentukan asrama Hogwarts yang sesuai berdasarkan sifat yang dimiliki seseorang dengan cara mencocokkannya dengan karakteristik masing-masing asrama.

Pada makalah ini, digunakan algoritma *string matching* dalam mengimplementasikan sebuah program untuk menentukan asrama Hogwarts yang sesuai. Misalnya terdapat sebuah string berisi sifat-sifat yang menggambarkan suatu asrama dan sebuah string berisi sifat seseorang. Dengan algoritma *string matching*, dapat diperiksa apakah orang tersebut cocok untuk masuk ke dalam asrama tersebut dengan cara menghitung jumlah sifat yang identik. Pencocokan string akan dilakukan untuk setiap sifat seseorang terhadap sifat-sifat yang menggambarkan asrama. Semakin banyak sifat yang identik, semakin cocok seseorang untuk masuk ke dalam asrama tersebut.

Pada implementasi program, digunakan algoritma Boyer-Moore dalam melakukan pencocokan string. Seperti yang telah dipaparkan sebelumnya, algoritma Boyer-Moore akan melakukan pencocokan string dengan cara mencocokkan karakter per karakter *pattern*, mulai dari kanan ke kiri, dengan karakter pada teks yang bersesuaian. Algoritma ini akan mengembalikan indeks dimana *pattern* ditemukan pada teks, atau -1 jika *pattern* tidak ditemukan pada teks.

Data sifat-sifat yang menggambarkan masing-masing asrama yang akan digunakan pada implementasi ini merupakan data yang diambil dari beberapa referensi internet dan dengan jumlah yang terbatas. Berikut merupakan data sifat-sifat yang dimaksud.

Asrama	Sifat
Gryffindor	berani, pemberani, suka menolong, penolong, sopan, tidak mengenal takut, kreatif, berjiwa kepemimpinan, pemimpin, dapat dipercaya, aktif, berwibawa, supel, ringan tangan, ceroboh, bersifat ksatria
Hufflepuff	pekerja keras, sabar, setia, loyal, adil, ramah, dermawan, murah hati, dewasa, baik, penyayang, berkarisma, karismatik, disukai, penasaran, kepo, disegani, berdedikasi, toleransi, tenggang rasa
Ravenclaw	pintar, cerdas, bijaksana, senang belajar, pembelajar, kreatif, ramah, friendly, penyayang, senang mencoba hal baru, kutu buku, berimajinasi tinggi, terorganisir, planner, penasaran, imajinasi tinggi, imajinatif, hangat, penuh pemikiran, senang belajar, rajin, individualistis
Slytherin	berambisi, ambisius, cerdik, licik, berani, pemberani, pemimpin, berjiwa kepemimpinan, banyak akal, jujur, suka

	menolong, keras kepala, bandel, gigih, pemikir, selalu ingin menang, ide cemerlang, straightforward, blakblakan, pintar, cerdas
--	---

Tabel 1 Karakteristik Asrama Hogwarts
(Sumber: Dokumen Pribadi)

Alur dari program yang telah dibuat adalah sebagai berikut.

1. Pengguna memberi masukan berupa sifat-sifat yang ia miliki dimana masing-masing sifat dipisahkan dengan ','.
2. Masukan sifat-sifat tersebut akan disimpan dalam suatu array, dengan terlebih dahulu dipisahkan berdasarkan delimiter tanda koma (,).
3. Untuk setiap sifat pengguna, dilakukan pencocokan string dengan sifat masing-masing asrama yang telah disimpan dalam variabel. Apabila ditemukan kecocokan, counter untuk asrama tersebut akan ditambah dengan satu.
4. Setelah semua sifat pengguna diperiksa, akan dicari asrama yang memiliki kecocokan sifat terbanyak dengan pengguna.
5. Kemudian, asrama yang cocok dengan pengguna akan ditampilkan pada layar.

Berikut merupakan pseudocode dari program utama yang dibuat.

Deklarasi

count, indices : array of integer

houses, houseCharacteristics, arrPersonalities : array of string

personalities, personality : string

maxCount, index : integer

Algoritma

count ← [0, 0, 0, 0]

houses ← ["Gryffindor", "Hufflepuff", "Ravenclaw", "Slytherin"]

{ Deklarasi sifat masing-masing asrama }

houseCharacteristics ← ["berani, pemberani, suka menolong, penolong, sopan, tidak mengenal takut, kreatif, berjiwa kepemimpinan, pemimpin, dapat dipercaya, aktif, berwibawa, supel, ringan tangan, ceroboh, bersifat ksatria",

"pekerja keras, sabar, setia, loyal, adil, ramah, dermawan, murah hati, dewasa, baik, penyayang, berkarisma, karismatik, disukai, penasaran, kepo, disegani, berdedikasi, toleransi, tenggang rasa",

"pintar, cerdas, bijaksana, senang belajar, pembelajar, kreatif, ramah, friendly, penyayang, senang mencoba hal baru, kutu buku, berimajinasi tinggi, terorganisir, planner, penasaran, imajinasi tinggi, imajinatif, hangat, penuh pemikiran, senang belajar, rajin, individualistis",

"berambisi, ambisius, cerdik, licik, berani, pemberani,

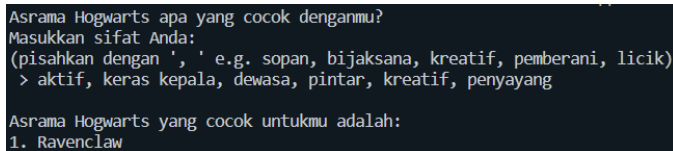
pemimpin, berjiwa kepemimpinan, banyak akal, jujur, suka menolong, keras kepala, bandel, gigih, pemikir, selalu ingin menang, ide cemerlang, straightforward, blakblakan, pintar, cerdas"]

```
input(personalities)
arrPersonalities ← personalities.split(", ")
for personality in arrPersonalities do
  for i ← 0 to 3 do
    if boyerMoore(characteristics[i], personality) ≠ -1 then
      count[i] ← count[i] + 1
    endif
  endfor
endfor

maxCount ← max(count)
indices ← [i for i, x in enumerate(count) if x = maxCount]
for index in indices do
  output(houses[index])
endfor
```

Dilakukan beberapa eksperimen untuk melakukan pengujian terhadap program ini.

1. Uji coba 1



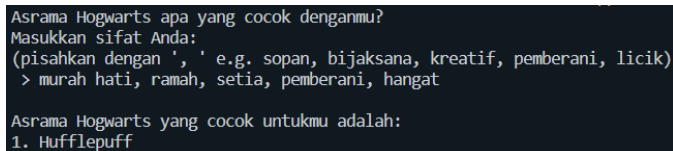
```
Asrama Hogwarts apa yang cocok denganmu?
Masukkan sifat Anda:
(pisahkan dengan ', ' e.g. sopan, bijaksana, kreatif, pemberani, licik)
> aktif, keras kepala, dewasa, pintar, kreatif, penyayang

Asrama Hogwarts yang cocok untukmu adalah:
1. Ravenclaw
```

Gambar 10 Uji Coba 1
(Sumber: Dokumen Pribadi)

Pada uji coba dengan masukan sifat aktif, keras kepala, dewasa, pintar, kreatif, penyayang, terdapat sebanyak 2 karakteristik asrama Gryffindor, 2 karakteristik asrama Hufflepuff, 3 karakteristik asrama Ravenclaw, dan 2 karakteristik asrama Slytherin. Oleh karena itu, diperoleh keluaran berupa asrama Ravenclaw.

2. Uji coba 2



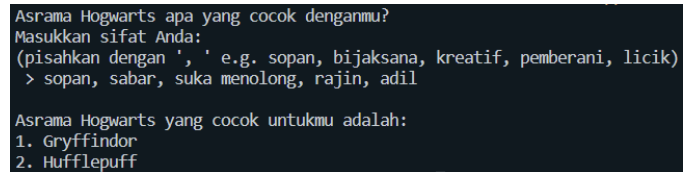
```
Asrama Hogwarts apa yang cocok denganmu?
Masukkan sifat Anda:
(pisahkan dengan ', ' e.g. sopan, bijaksana, kreatif, pemberani, licik)
> murah hati, ramah, setia, pemberani, hangat

Asrama Hogwarts yang cocok untukmu adalah:
1. Hufflepuff
```

Gambar 11 Uji Coba 2
(Sumber: Dokumen Pribadi)

Pada uji coba dengan masukan sifat murah hati, ramah, setia, pemberani, hangat, terdapat sebanyak 1 karakteristik asrama Gryffindor, 3 karakteristik asrama Hufflepuff, 2 karakteristik asrama Ravenclaw, dan 1 karakteristik asrama Slytherin. Oleh karena itu, diperoleh keluaran berupa asrama Hufflepuff.

3. Uji coba 3



```
Asrama Hogwarts apa yang cocok denganmu?
Masukkan sifat Anda:
(pisahkan dengan ', ' e.g. sopan, bijaksana, kreatif, pemberani, licik)
> sopan, sabar, suka menolong, rajin, adil

Asrama Hogwarts yang cocok untukmu adalah:
1. Gryffindor
2. Hufflepuff
```

Gambar 12 Uji Coba 3
(Sumber: Dokumen Pribadi)

Pada uji coba dengan masukan sifat sopan, sabar, suka menolong, rajin, adil, terdapat sebanyak 2 karakteristik asrama Gryffindor, 2 karakteristik asrama Hufflepuff, 1 karakteristik asrama Ravenclaw, dan 1 karakteristik asrama Slytherin. Oleh karena itu, diperoleh keluaran berupa asrama Gryffindor dan Hufflepuff.

V. KESIMPULAN

Algoritma *string matching* merupakan algoritma yang dapat digunakan untuk menyelesaikan berbagai persoalan, salah satunya adalah untuk menentukan asrama Hogwarts yang sesuai berdasarkan sifat yang dimiliki. Terdapat tiga algoritma yang populer dalam melakukan *string matching*, yaitu algoritma Brute Force, Knuth-Morris-Pratt, dan Boyer-Moore.

Dalam makalah ini, digunakan algoritma Boyer-Moore dalam mengimplementasikan program yang dibuat. Program mampu menerima masukan berupa sifat-sifat pengguna dan menampilkan asrama yang sesuai berdasarkan masukan tersebut. Namun, program ini masih dapat dikembangkan, salah satunya dengan cara mengolah masukan terlebih dahulu sebelum diperiksa agar hasil yang diperoleh lebih akurat.

VIDEO LINK YOUTUBE

<https://youtu.be/ww0QVY3wcLU>

UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas berkat yang telah diberikan sehingga penulis dapat menyelesaikan makalah yang berjudul “Aplikasi String Matching dalam Menentukan Asrama Hogwarts yang Sesuai” ini dengan baik dan tepat waktu. Penulis juga mengucapkan terima kasih kepada orang tua, kakak, keluarga, dan teman yang selalu mendukung penulis.

Tak lupa, penulis juga mengucapkan terima kasih kepada seluruh dosen mata kuliah IF2211 Strategi Algoritma, terutama Bapak Rinaldi Munir selaku Dosen Strategi Algoritma Kelas 3 atas ilmu yang telah diberikan sehingga dapat membantu penulis dalam pembuatan makalah ini.

REFERENCES

- [1] Gresge, Gabrielle. *Does Your Myers-Briggs Type Align With Your Harry Potter House?*. Diakses pada 7 Mei 2022, dari <https://www.brit.co/myers-briggs-type-align-harry-potter-house/>.
- [2] Harry Potter Wiki Indonesia. *Asrama Hogwarts*. Diakses pada 14 Mei 2022, dari <https://personalitygrowth.com/myers-briggs-harry-potter-hogwarts-houses/>.
- [3] Harry Potter Wiki. *Hogwarts Houses*. Diakses pada 14 Mei 2022, dari <https://personalitygrowth.com/myers-briggs-harry-potter-hogwarts-houses/>.
- [4] Munir, Rinaldi. *Pencocokan String (String/Pattern Matching)*. Diakses pada 8 Mei 2022, dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>.
- [5] Stafford, Scott. *Myers Briggs: Harry Potter Hogwarts Houses*. Diakses pada 7 Mei 2022, dari <https://personalitygrowth.com/myers-briggs-harry-potter-hogwarts-houses/>.
- [6] Torn, Simone. *Myers-Briggs® Types Sorted Into Their Hogwarts Houses*. Diakses pada 7 Mei 2022, dari <https://screenrant.com/myers-briggs-types-sorted-hogwarts-houses/>.

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, 23 Mei 2022



Flavia Beatrix Leoni A. S. - 13520051