

# Penerapan Algoritma Pattern Matching pada Scrabble Game

Kharisma Nugrahandani Restuti (13512601)

*Program Studi Teknik Informatika*

*Sekolah Teknik Elektro dan Informatika*

*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*

*<sup>1</sup>13512601@std.stei.itb.ac.id*

**Abstract**—Permainan Scrabble merupakan permainan klasik yang mengkombinasikan kotak-kotak huruf untuk mendapatkan kata-kata yang sesuai standar kamus dalam waktu tertentu untuk mendapatkan poin terbanyak. Permainan ini bertujuan untuk meningkatkan kemampuan otak dalam mengkombinasikan huruf-huruf menjadi kata secara cepat dan tepat dan untuk memperluas kosa kata dalam bahasa yang ditentukan dalam permainan. Ketika menentukan kesesuaian kombinasi huruf dengan kosa kata yang benar sesuai kamus diperlukan algoritma pattern matching serta penerapannya dari permainan klasik ke dalam suatu program aplikasi, yang akan dibahas dalam makalah ini.

**Kata Kunci**—Pattern Matching, Scrabble, Kosa Kata, Kamus.

## I. PENDAHULUAN

Scrabble game merupakan permainan penyusunan huruf menjadi suatu kata baku. Pemain berusaha menyusun kata yang sesuai dari kotak-kotak huruf yang tersedia baik mendatar maupun menurun seperti teka-teki silang. Kata-kata yang dibuat merupakan kata-kata yang baku dan sesuai standar kamus dalam bahasa yang dimainkan.

Pada awal perkembangannya, scrabble merupakan permainan klasik yang dimainkan oleh 2 hingga 4 orang. Pemain menyusun huruf-huruf tersebut atas papan, sehingga pola kata-kata terbentuk seperti pola pada teka-teki silang. Namun pada perkembangannya, banyak jenis-jenis permainan scrabble yang bisa dimainkan satu orang melalui berbagai *platform*.

Permainan scrabble memerlukan kemampuan verbal dalam membentuk kata dengan lancar, karena *game* ini membutuhkan kecepatan dalam pengambilan kata-kata yang tepat dalam memori. Kemudian kemampuan visuospatial, karena tata ruang kata dan huruf pada kotak tertentu di papan menentukan nilai poin bermain, dan kemampuan numerik, karena pemain perlu menghitung probabilitas dan kecepatan dalam menggunakan sifat numerik yang terkait dengan berbagai kombinasi kata-kata yang terletak di tempat yang berbeda di papan permainan.

Dalam Scrabble, pemain meletakkan kotak-kotak huruf di horisontal (kiri ke kanan) atau vertikal (atas ke bawah) secara berurutan untuk membuat sebuah kata yang sah.

Seperti catur dan permainan lainnya, Scrabble dimainkan dalam kompetisi internasional, nasional, dan regional untuk uang dan ketenaran. Scrabble tidak hanya permainan pengetahuan kata, tetapi juga permainan matematika dan probabilitas.

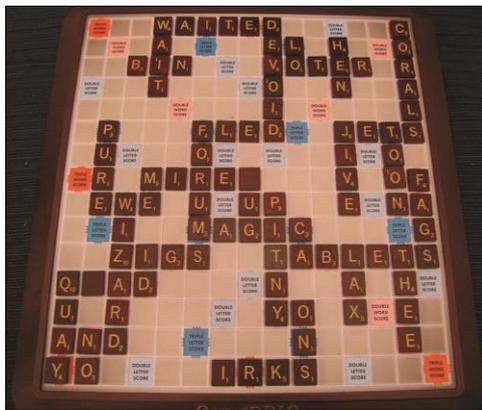
Pemain ahli perlu memperkirakan probabilitas memilih huruf-huruf yang tersisa untuk dimainkan dengan cepat untuk kata-kata yang berbeda di papan untuk menentukan beberapa kemungkinan kata akan menghasilkan nilai poin tertinggi. Scrabble kompetitif adalah permainan matematika "(Brown, 2006, hal. M4). Pemain ahli biasanya mendedikasikan sejumlah besar waktu untuk mempelajari 180.000 kata yang tercantum dalam The Tournament Resmi dan Klub Daftar Kata.

## II. DASAR TEORI

### 2.1 Sejarah Scrabble

Alfred Mosher Butts, seorang arsitek pengangguran yang memutuskan untuk menciptakan sebuah papan permainan. Tidak seperti permainan yang lebih tua, seperti catur, Go, atau backgammon, Scrabble dimiliki oleh perusahaan permainan (Hasbro, Inc) dan oleh karena itu tidak dapat disalin atau disebutkan tanpa merek dagang. Papan catur memiliki 64 kotak dan mencakup 32 buah, dimana 6 memiliki fungsi yang berbeda. Sebagai perbandingan, Scrabble memiliki 225 kotak dan 100 ubin, yang masing-masing dari 26 huruf abjad dalam kata-kata berbahasa Inggris.

Huruf memiliki nilai poin sesuai dengan kerangkanya. Misalnya, huruf z bernilai 10 poin dan muncul hanya sekali, sedangkan huruf a dan i masing-masing senilai 1 poin dan muncul sembilan kali masing-masing.



Gambar 1. Scrabble board saat permainan berakhir

Butts melakukan riset pasar dan menyimpulkan bahwa permainan jatuh ke dalam tiga kategori: permainan angka, seperti dadu dan bingo, memindahkan permainan, seperti catur, dan permainan kata-kata, seperti anagram. Butts ingin menciptakan sebuah permainan yang menggabungkan keterampilan kosa kata teka-teki silang dan anagram, dengan tambahan unsur kebetulan. Permainan ini awalnya bernama Lexico, tapi Butts akhirnya memutuskan untuk memanggil permainan "Criss-Cross Words."

Butts mempelajari halaman depan The New York Times untuk menghitung seberapa sering 26 huruf dari bahasa Inggris digunakan. Ia menemukan bahwa huruf vokal jauh lebih sering digunakan daripada huruf konsonan, dengan E yang paling sering digunakan dari huruf vokal. Setelah mencari tahu frekuensi penggunaannya, Butts berusaha mencari nilai poin yang berbeda untuk setiap huruf dan memutuskan berapa banyak dari setiap huruf yang akan dimasukkan dalam permainan. Huruf S menimbulkan masalah, karena sering digunakan untuk penggunaan kata jamak, sehingga Butts memutuskan untuk memasukkan hanya empat huruf S dalam permainan untuk membatasi penggunaan bentuk jamak.

## 2.2 Algoritma Pattern Matching

Ada beberapa algoritma yang digunakan dalam makalah ini, antara lain:

### 2.2.1 KMP

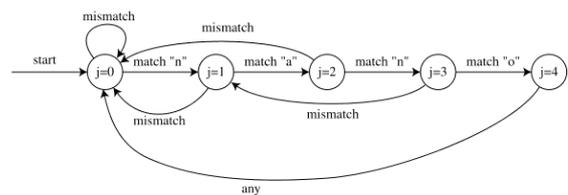
Algoritma Knuth-Morris-Pratt (KMP) dikenal sebagai algoritma pencocokan yang tepat untuk waktu yang linear. KMP mengecek dari kiri ke kanan dan melakukan pergeseran lebih dari 1 karakter. Pada Algoritma KMP, informasi disimpan guna melakukan jumlah pergeseran. Misalkan kita sedang mencari pola "nano" dalam teks "banananobano".

```

0 1 2 3 4 5 6 7 8 9 10 11
T: b a n a n a n o b a n o
i=0: X
i=1:  X
i=2:   n a n X
i=3:    X
i=4:     n a n o
i=5:      X
i=6:       n X
i=7:        X
i=8:         X
i=9:          n X
i=10:           X

```

Dari contoh tersebut, string "nano" akan mulai diperiksa dari karakter keempat jika mismatch terjadi, karena "nan" panjangnya tiga, maka tiga karakter dilewati, dan *pattern* mulai diperiksa dari karakter keempat.



Gambar 2. Finite Automata pattern "nano"

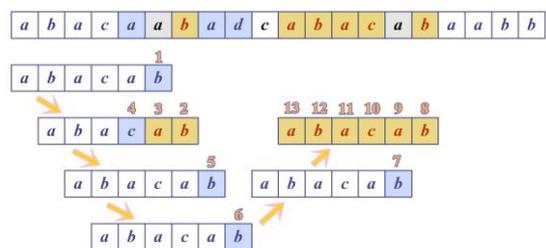
### 2.2.2 Boyer Moore

Algoritma Boyer dan Moore membandingkan *pattern* dengan *text* dari kanan ke kiri. Jika karakter pada *text* yang dibandingkan dengan karakter paling kanan pada *pattern* tidak sesuai, maka *pattern* digeser dengan posisi m di belakang karakter *text*. Contoh dari *pattern* "abacab". Karakter 'b' terletak pada kejadian terakhir yaitu 6 karena terakhir 'b' menempati indeks ke-6. Untuk karakter yang tidak ada dalam *pattern* akan bernilai -1 sebagai indeksnya.

	A	b	a	c	a	b
P	1	2	3	4	5	6

X	a	b	c	d
L(x)	5	6	4	-1

Tabel 2.2 Kejadian pada *pattern* "abacab"



Gambar 2.2 Contoh implementasi algoritma boyer moore

### III. ANALISIS DAN PEMBAHASAN

#### 3.1 Mekanisme Dasar Permainan Scrabble

Dalam scrabble, pemain mendapatkan poin dengan membangun kata-kata dengan kotak-kota huruf dan menempatkannya pada papan. Setiap huruf memiliki poin nilai yang berbeda, ada 100 kotak. Elemen-elemen yang diperlukan untuk permainan scrabble klasik:

1. Dua sampai empat pemain.
2. Kantong kotak-kotak Scrabble.
3. Papan scrabble.
4. Pena dan kertas.

Langkah-langkah permainan scrabble klasik sebagai berikut :

1. Empat pemain mengeluarkan masing-masing satu kotak huruf dari kantong kotak-kotak scrabble. Pemain yang mendapat huruf yang paling dekat dengan huruf "A" akan bermain dulu. Setiap pemain kemudian mengambil tujuh kotak dari kantong kotak. Pemain pertama memulai permainan dengan menempatkan sebuah kata pada simbol bintang di tengah papan, baik horisontal maupun vertikal. Bintang ini menandakan kata yang dibentuk pemain memiliki *double score*.
2. Hitung setiap skor kata dengan menambahkan poin nilai untuk setiap huruf dalam kata. Poin nilai berkisar dari 0 poin untuk kotak kosong hingga 10 poin untuk huruf "Q" dan "Z". Penentuan poin bisa dilihat dalam tabel 1. Gunakan kertas dan pena untuk mencatat skor kata setiap pemain. Beberapa kotak di papan memungkinkan pemain untuk melipatgandakan nilai dua atau tiga kali.
3. Lanjutkan permainan berlawanan dengan arah jarum jam. Pada setiap giliran, pemain dapat memutuskan untuk menempatkan sebuah kata di atas papan untuk mengganti kotak, atau melewatkannya. Seorang pemain dapat memutuskan untuk bertukar kotak, dan menggantinya dengan jumlah setara dari dalam kantong kotak.
4. Setelah menempatkan kata di papan permainan, ambil jumlah kotak baru yang sama dari kantong kotak sehingga setiap pemain selalu memiliki tujuh kotak.
5. Lanjutkan permainan. Semua huruf yang dimainkan harus menyentuh setidaknya satu huruf yang sudah ada di papan untuk membentuk setidaknya satu kata yang lengkap. Ketika salah satu pemain telah menggunakan semua kotak atau tidak ada pemain yang bisa membuat kata-kata lagi, permainan berakhir.

Jika seorang pemain menggunakan semua tujuh kotak mereka pada satu giliran, mereka menambahkan bonus 50 poin untuk skor kata-kata mereka.

6. Lihat kamus hanya jika pemain ingin menantang kata-kata yang dibuat pemain lain. Jika kata tersebut belum ada di kamus, maka pemain kehilangan giliran mereka. Jika kata tersebut diperbolehkan, penantang kehilangan giliran mereka.
7. Hitung skor akhir masing-masing pemain dengan menjumlahkan poin dari semua nilai kata. Kemudian, kurangi poin dengan jumlah huruf yang tidak dimainkan oleh mereka.

Jika salah satu pemain telah menggunakan semua huruf mereka, tambahkan jumlah huruf yang tidak dimainkan oleh pemain-pemain lainnya untuk menambah skor. Pemain dengan skor tertinggi adalah yang menang permainan.

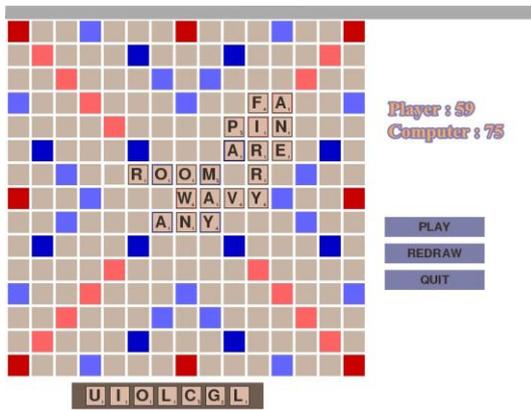
A	1	J	8	S	1
B	3	K	5	T	1
C	3	L	1	U	1
D	2	M	3	V	4
E	1	N	1	W	4
F	4	O	1	X	8
G	2	P	3	Y	4
H	4	Q	10	Z	10
I	1	R	1		0

Tabel 1. Daftar poin nilai setiap huruf

#### 3.2 Implementasi permainan Scrabble pada komputer

Permainan Scrabble membutuhkan kemampuan dalam mengkombinasikan kosakata, strategi, pengenalan *pattern* serta keberuntungan. Namun, permainan Scrabble di atas papan, memungkinkan terjadinya beberapa masalah, diantaranya kotak-kotak huruf yang mungkin hilang, atau perhitungan poin nilai yang rumit dan kompleks untuk setiap kata yang dibentuk, serta pencocokan kata yang dibentuk sesuai pada standar yang diijinkan atau sesuai pada kamus.

Scrabble dalam platform komputer ini dimainkan di atas papan untuk penempatan kotak-kotak huruf yang mungkin bisa dibentuk, seperti yang diilustrasikan pada gambar berikut.



Gambar 2. Permainan Scrabble pada komputer

Dalam pembuatannya terdapat satu set *slot* di mana semua kotak-kotak huruf dapat diletakkan dan dipindahkan.

Untuk masing-masing *slot*, dapat dilakukan percobaan untuk menempatkan semua permutasi dari kotak-kotak huruf yang tersedia. Setiap kotak yang kosong berarti mencoba 26 kemungkinan huruf.

Untuk setiap permainan, harus memeriksa apakah semua kata terbentuk secara valid, kemudian menghitung poin nilainya. Hal ini membutuhkan strategi analisa pencarian string.

Ada beberapa masalah dengan pendekatan ini, yaitu adalah pencarian yang sangat lambat. Permainan yang mempertimbangkan beberapa ribu kemungkinan *slot*, untuk masing-masing terdapat lebih dari 5000 permutasi dan lebih dari 3 juta jika mempertimbangkan terdapat slot kosong. Jika dijalankan dengan cara brute-force dapat berlangsung selama beberapa jam atau lebih pada komputer modern.

Penilaian skornya sangat kompleks, sehingga hanya kata-kata yang valid yang dihitung. Kemudian sebelum menempatkan kotak huruf 'blank' atau kosong, maka perlu dicari tahu terlebih dahulu apakah terdapat kata-kata yang sesuai dengan adanya kotak kosong tersebut. X\_YGYK tidak memiliki arti apa-apa, bukan kata yang valid, sehingga tidak dapat terhitung.

Karena kosong memiliki nol poin, maka perlu mencari tahu apakah salah satu hurufnya dalam baris kata tersebut valid.

Algoritma untuk menghitung kemungkinan *slot* dapat menghasilkan kata yang duplikat, sehingga perlu memastikan setiap slot unik dapat memotong sebesar 30% hingga 50% biaya pengolahan.

Semua optimasi algoritmik masih bisa dikatakan benar dan akan tetap selalu menghasilkan skor kata terbaik, tidak ada pengecualian. Meski begitu, karena algoritma tergantung pada jumlah kemungkinan kotak huruf (dan kotak kosong), mengingat setiap opsi yang memungkinkan dapat

memakan waktu yang lama dalam kasus yang ekstrim.

Beberapa contoh kata yang terdapat pada kamus yang dapat dipakai dalam permainan Scrabble ini di antara terdapat pada table berikut

7 letter words
goliath[11], alright[11]
6 letter words
mohair[11], thairm[11], thiram[11], alight[10], aright[10], righto[10], glioma[9], glamor[9], thoria[9], harlot[9], liroth[9], maloti[8], mitral[8], ramtil[8], mortal[8], gloria[7], galiot[7], latigo[7], rialto[6], tailor[6]
5 letter words
ogham[11], might[11], ihram[10], tharm[10], mirth[10], laigh[9], light[9], garth[9], girth[9], grith[9], right[9], amigo[8], imago[8], gloam[8], nilar[8], horal[8], laith[8], lathi[8], altho[8], loath[8], lotah[8], litho[8], thiol[8], magot[8], airth[8], thirl[8], torah[8], moira[7], molar[7], moral[7], amort[7], logia[6], argil[6], glair[6], grail[6], algor[6], argol[6], goral[6], largo[6], gloat[6], tragi[6], argot[6], gator[6], groat[6], griot[6], trigo[6], trail[5], trial[5], ratio[5], tolar[5], lirot[5], triol[5]
4 letter words
halm[9], holm[9], harm[9], math[9], moth[9], ghat[8], goth[8], hail[7], hila[7], magi[7], glam[7], glim[7], ohia[7], halo[7], ogam[7], glom[7], hair[7], harl[7], gram[7], grim[7], hoar[7], hora[7], gorm[7], halt[7], lath[7], hilt[7], oath[7], thio[7], holt[7], loth[7], hart[7], rath[7], tahr[7], thir[7], thro[7], lima[6], mail[6], loam[6], mola[6], limo[6], milo[6], moil[6], amir[6], mair[6], rami[6], marl[6], mora[6], roam[6], malt[6], milt[6], atom[6], moat[6], omit[6], molt[6], mart[6], tram[6], trim[6], mort[6], glia[5], agio[5], gaol[5], goal[5], ragi[5], girl[5], giro[5], gait[5], gilt[5], goat[5], toga[5], grat[5], girt[5], grit[5], trig[5], grot[5], trog[5], aril[4], lair[4], lari[4], liar[4], lira[4], rail[4], rial[4], oral[4], roil[4], alit[4], lati[4], tail[4], tali[4], iota[4], alto[4], lota[4], tola[4], loti[4], toil[4], airt[4], tirl[4], rato[4], rota[4], taro[4], tora[4], riot[4], roti[4], tiro[4], tori[4], trio[4], rotl[4]
3 letter words
ham[8], him[8], hag[7], ghi[7], ahi[6], gam[6], mag[6], mig[6], hao[6], aim[5], ami[5], lam[5], mil[5], gal[4], lag[4], ago[4], goa[4], ail[3]

Tabel 2. Contoh kata dan nilai poin untuk permainan Scrabble

### 3.3 Analisa Algoritma pada permainan Scrabble

Database kata-kata dapat digunakan dalam

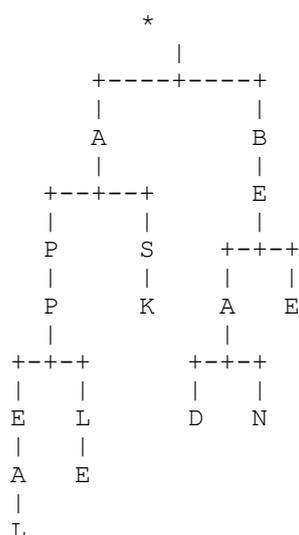
penyelesaian algoritma untuk permainan Scrabble ini. Kemudian setelah terbentuk kata-kata atau *pattern*, dapat mulai menemukan kata-kata untuk ditempatkan di dalamnya.

Misal:

app → lapp → leap → lecapp → ... → lappe → eappc → ... → appl → apple (jawaban yang benar)

"app" adalah awal dari kata dan dapat membatasi pencarian. Namun tidak setiap kata yang memiliki "app" di dalamnya, sehingga lebih mudah untuk menemukan kata-kata dengan huruf pada posisi yang dikenal daripada untuk mengevaluasi huruf-huruf pada posisi sembarang di dalam kata. Jika telah diketahui memiliki 5 huruf dalam kata tersebut dimulai dengan "app", ini cukup mudah digunakan.

Pembentukan *pattern* dapat digambar dalam pohon sebagai berikut :



Dalam pohon di atas terdapat kata-kata *appeal*, *apple*, *ask*, *bead*, *bean*, *be*, dan *bee*. Bila Anda menemukan huruf yang tidak diketahui, gunakan wildcard (yaitu, memilih semua anak dan menurun pada semua *path*).

Jika ingin menggunakan huruf P sebagai huruf kedua, maka dapat dimulai dari simpul akar dan mengambil setiap cabang (yang akan menjadi setiap huruf dalam alfabet, dengan asumsi bahwa itu terdapat pada kamus, sehingga merupakan kata-kata yang valid), kemudian menuju cabang "P" dan kemudian melanjutkan cabangnya atau berhenti pada huruf yang tepat.

Ada beberapa analisis untuk menentukan algoritma yang efektif berdasarkan kompleksitasnya, metode algoritma, dan sebagainya. Algoritma brute-force berjalan dalam waktu  $O(mn)$  untuk kasus terburuknya, hal ini berarti bahwa semua karakter dalam *pattern* harus dibandingkan dengan setiap

karakter tunggal dalam *text*. Namun, sebagian besar pencarian dalam *text* normal biasanya berlangsung dalam waktu  $O(m+n)$  yang sangat cepat.

Algoritma KMP biasanya berjalan dalam waktu  $O(m+n)$ , dan sangat cepat. Namun besar kemungkinan algoritma KMP untuk tidak sesuai untuk ukuran karakter dalam *text* yang besar. Namun untuk permainan Scrabble, algoritma KMP cukup baik untuk digunakan.

## VI. KESIMPULAN

Pattern matching merupakan algoritma yang sesuai dalam menentukan kevalidan *pattern* pada suatu *text*, atau dalam hal ini adalah kamus yang digunakan untuk permainan Scrabble.

## REFERENCES

- [1] "Pattern matching and text compression algorithms," Maxime Crochemore. France.
- [2] <http://scienceblog.com/47203/super-scrabble-players-push-brain-ability-beyond-what-was-thought-possible/>
- [3] Munir, Rinaldi. 2009. *Diktat Kuliah IF3051 Strategi Algoritma*. Program Studi Teknik Informatika STEI ITB
- [4]
- [5] Hargreaves IS et al (2011). How a hobby can shape cognition: visual word recognition in competitive Scrabble players. *Memory & Cognition*. DOI 10.3758/s13421-011-0137-5
- [6] [http://www.hasbro.com/scrabble/en\\_US/butts.cfm](http://www.hasbro.com/scrabble/en_US/butts.cfm)
- [7] <http://stackoverflow.com/questions/2497986/writing-an-algorithm-for-scrabble>
- [8] <http://www.youtube.com/watch?v=swlg3vQXboE>
- [9] <http://www.ics.uci.edu/~eppstein/161/960227.html>

## 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, 20 Desember 2013



Ttd

Kharisma Nugrahandani Restuti (13512601)